目 录

序言

前言

预备实验 MATLAB 基础 …… 1

§0.1 MATLAB 入门 …… 1

§0.2 矩阵和数组运算 …… 4

§0.3 图形 …… 8

§0.4 程序设计 …… 12

§0.5 在线帮助和文件管理 …… 16

§0.6* 符号数学工具箱 …… 17

§0.7 习题 …… 19

实验一 投入产出平衡 矩阵和线性方程组 …… 20

§1.1 引例:国民经济投入产出综合平衡 …… 20

§1.2 数学理论复习:线性代数 …… 21

§1.3 线性代数运算 MATLAB 命令 …… 22

§1.4 实验例题 …… 26

§1.5 实验习题 …… 28

§1.6 补充知识:Gauss 消去法和病态方程组 …… 30

实验二 购房贷款的利率 非线性方程和迭代 …… 34

§2.1 引例:购房贷款的利率 …… 34

§2.2 数学理论复习:非线性方程(组) …… 35

§2.3 数值解法:图解法和迭代法 …… 35

§2.4 解方程和方程组的 MATLAB 命令 …… 37

§2.5 实验例题 …… 38

§2.6 实验习题 …… 39

§2.7 补充知识:认识混沌 …… 40

实验三 最佳订货量 极限、导数和极值 …… 45

§3.1 引例:最佳订货量问题 …… 45

§3.2 数学理论复习:微分学 …… 46

§3.3 数值微分 …… 48

§3.4 求极限、导数和极值的 MATLAB 命令 …… 48

§3.5 实验例题 …… 50

§3.6 实验习题 …… 52

§3.7 补充知识:计算的局限性 …… 54

实验四 数学家的生日蛋糕 积分 …… 57
§4.1 引例:数学家的生日蛋糕 …… 57
§4.2 数学理论复习:积分 …… 58
§4.3 数值积分:梯形法和重积分 …… 59
§4.4 求积分的 MATLAB 命令 …… 61
§4.5 实验例题 …… 63
§4.6 实验习题 …… 65
§4.7 补充知识:变步长积分法 …… 66
实验五 导弹系统的改进 微分方程 …… 69
§5.1 引例:导弹系统的改进 …… 69
§5.2 数学理论复习:常微分方程 …… 70
§5.3 微分方程数值解法:Euler 法 …… 71
§5.4 解微分方程的 MATLAB 命令 …… 72
§5.5 实验例题 …… 74
§5.6 实验习题 …… 76
§5.7 补充知识:稳定性 …… 77
实验六 零件参数设计 随机模拟 …… 81
§6.1 引例:零件参数设计 …… 81
§6.2 数学理论复习:概率论 …… 81
§6.3 随机模拟原理 …… 83
§6.4 数据分析和随机数生成的 MATLAB 命令 …… 84
§6.5 实验例题 …… 87
§6.6 实验习题 …… 90
§6.7 补充知识:小概率陷阱 …… 90
实验七 身高、体重与体育成绩 统计推断 …… 94
§7.1 引例:学生的身高、体重与体育成绩 …… 94
§7.2 数学理论复习:数理统计的基本概念 …… 95
§7.3 MATLAB 统计分析工具箱 …… 96
§7.4 统计推断方法 …… 97
§7.5 实验例题 …… 101
§7.6 实验习题 …… 104
§7.7 补充知识:非线性回归 …… 106
实验八 凸轮设计 插值与拟合 …… 114
§8.1 引例:万能拉拔机凸轮设计 …… 114
§8.2 理论基础:数据插值和拟合 …… 115
§8.3 数据拟合 MATLAB 命令 …… 116
§8.4 实验例题 …… 120
§8.5 实验习题 …… 124
§8.6 补充知识:样条函数工具箱和二元插值 …… 125

实验九 最佳连续投资方案 线性规划 …… 131
§9.1 引例:最佳连续投资方案 …… 131
§9.2 线性规划基本理论复习 …… 132
§9.3 求解线性规划的 MATLAB 命令 …… 134
§9.4 实验例题 …… 135
§9.5 实验习题 …… 137
§9.6 补充知识:单纯形算法 …… 138
实验十 飞行管理问题 非线性规划 …… 146
§10.1 引例:飞行管理问题 …… 146
§10.2 非线性规划基本理论复习 …… 148
§10.3 求解非线性规划的 MATLAB 命令 …… 150
§10.4 实验例题 …… 152
§10.5 实验习题 …… 155
§10.6 补充知识:非线性规划算法 …… 156
实验十一 货车装货方案 整数规划 …… 158
§11.1 引例:货车装货方案 …… 158
§11.2 整数线性规划基本理论复习 …… 159
§11.3 整数线性规划分枝定界法 MATLAB 程序 …… 161
§11.4 0-1 型整数线性规划 …… 163
§11.5 0-1 型整数线性规划计算的 MATLAB 程序 …… 163
§11.6 实验例题 …… 165
§11.7 实验习题 …… 169
实验十二 通信联络网的建立 图与网络优化 …… 171
§12.1 引例:通信联络网的建立 …… 171
§12.2 图与网络的基本理论复习 …… 171
§12.3 Kruskal 算法与 Dijkstra 算法的 MATLAB 程序 …… 173
§12.4 实验例题 …… 175
§12.5 实验习题 …… 178
实验十三 生产计划的制定 动态规划 …… 180
§13.1 引例:生产计划的制定 …… 180
§13.2 动态规划的基本理论复习 …… 181
§13.3 动态规划逆序算法的 MATLAB 程序 …… 183
§13.4 实验例题 …… 185
§13.5 实验习题 …… 191
附录 A MATLAB 函数命令索引表 …… 193
附录 B MATLAB 工具箱函数命令索引表 …… 201
习题参考答案 …… 209
参考文献 …… 213

预备实验　MATLAB 基础

本预备实验从数学实验课程的需要出发，介绍 MATLAB 一些必要的基础知识。我们并不要求初学者能很快掌握所有内容，读者只要通过本预备实验，理解 MATLAB 的一些特征，为后继实验打下基础就足够了。其中较深的内容（特别是打 * 号部分）可留待以后慢慢消化。

§0.1　MATLAB 入门

启动 MATLAB 后，就进入 MATLAB 命令窗口（Command Window）或称工作空间（Workspace），见图 0-1。若你的 MATLAB 装在英文 Windows 中会出现提示符》，在提示符后键入任意合法命令，回车后 MATLAB 立即运算并显示结果。若 MATLAB 装在中文 Windows 中，这一提示符是看不见的，但它仍占据一定的位置。本书约定

(1) 所有在命令窗口输入的命令都用 Courier New 字体，并以》开头，请读者注意》为系统提示符，不要以为是输入字符。

(2) 显示结果用 Times New Roman 字体。

(3) %号后面的文字用于注释，并不参与运算，实验时也不必输入。

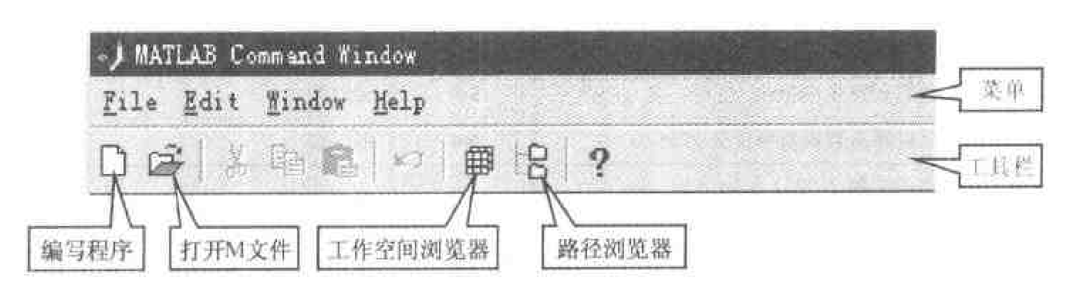


图 0-1　命令窗口菜单与工具栏

1. 简单的运算

看一个简单的例子，计算

$$V=\frac{4}{3}\pi r^3\text{，其中 } r=2$$

用 MATLAB 计算如下：

```
》r=2;                %分号“;”使此运算结果不显示
》v=4/3*pi*r^3        %没有分号，显示结果
v=
    33.5103           %系统直接显示结果，就像计算器那样
```

注：(1) MATLAB 命令书写格式灵活，可多命令写一行，也可一个命令写多行。同一行命令

用逗号或分号分开,若命令很长,一行不够,可用三点"..."续行。

(2) MATLAB 允许使用↑↓←→键来编辑前面的命令。试一试你就明白了。

2. 变量和数据

(1) 变量类型

MATLAB 使用变量最常用的有数值数组(double array)和字符串(char array)两类。所有数值变量以双精度(double)方式存储,不区分整数、实数、复数等,变量类型和数组大小也无需预先定义,例如

```
» a = 1 + 2* i
a=
    1.0000+2.0000i          %复数
» a1 = 'This is a string'
a1=
This is a string            %字符串
» A = [1   2;  -1   3]
A=
        1        2
       -1        3          %二维数组,即矩阵
```

(2) 常量和特殊变量

MATLAB 中一些常用的常量见表 0-1。

表 0-1 常量一览表

常量名	说明	常量名	说明
i 或 j	虚数单位 $\sqrt{-1}$	realmax	最大正实数 1.7977×10^{308}
pi	圆周率 $\pi=3.1415\cdots$	Inf	无穷大
eps	计算机浮点数识别精度 2.22×10^{-16}	NaN	不定值
realmin	最小正实数 2.2251×10^{-308}		

ans 是系统本身一个特殊变量名,若运算结果没有赋予任何变量,系统就自动将其赋予 ans。

(3) 变量查询和清除

MATLAB 命令窗口中用户变量可用 who 或 whos 查询,键入

```
» who
Your variables are:
A        a        a1        r        v
» whos
  Name      Size       Bytes    Class
   A        2x2         32      double array
   a        1x1         16      double array (complex)
   a1       1x16        32      char array
   r        1x1          8      double array
   v        1x1          8      double array
```

Grand total is 23 elements using 96 bytes

以上信息也可从工具栏中的工作空间浏览器(Workspace Browser)观察到。

变量的值可通过键入变量名得到,例如

```
» a                       %显示 a 的值
a=
    1.0000+2.0000i
```

若命令窗口中有些变量不再使用,可使用命令 clear 来清除。

```
» clear a A               %清除变量 a 和 A
» a
??? Undefined function or variable a.       %说明 a 已清除
» a1
a1=
This is a string          %a1 未清除
» clear                   %清除工作空间所有变量(慎用!)
```

注意 clear 与菜单 Edit\Clear session 的区别。后者作用是将稿纸(窗口显示)擦干净。

(4) 变量命名规则

MATLAB 变量名总以字母开头,由字符、数字和下划线组成。有效字符长度为 31 个,且区分大小写,如 a 与 A 表示不同变量。*用户在定义变量时,要尽量防止它与系统的常量名、特殊变量名、函数名等冲突*,系统不会告诉你冲突发生,造成的后果是系统有些原来的功能暂不能使用。当这些变量被清除或 MATLAB 重新启动后,这些功能得以恢复。

(5) 数据显示格式

MATLAB 缺省的数据显示格式为:当结果为整数,就作为整数显示;当结果是实数,以小数点后 4 位的精度显示。若结果的有效数字不在这一范围,以科学计数法显示(如 1e-6 表示 10^{-6})。数据显示格式可通过命令 format 改变。需要指出的是,显示格式的改变不会影响数据的实际值,所以不会影响计数精度。其计数精度约为 16 位有效数字。

```
» c=pi
c=
  3.1416
» format rational; c
c=
  355/113      %最接近的有理数之一
» format long; c
c=
  3.14159265358979  %小数点后 14 位
» format; c
c=
  3.1416          %恢复
```

MATLAB 还允许使用 fprintf 格式化输出,其使用方法与 C 语言基本一致。

```
» fprintf ('%20.6f', c)
```

```
        3.141593
```

(6) 数据保存和调用

当我们退出 MATLAB 时，命令窗口中变量不复存在。为了保留变量值，我们可在退出之前用命令 save 将变量连同它的值用二进制方式存储在数据文件中（详见 save 的帮助信息）。例如

```
》A=[1, 2; -1, 3];
》save                %所有变量和数据写入数据文件 matlab.mat
```

好了，现在退出 MATLAB，再次启动，用

```
》A
??? Undefined function or variable A.      %工作空间是空的
》load                %从 matlab.mat 调出
》A
A=
      1        2
     -1        3              %说明矩阵 A 已存在于命令窗口
```

菜单 File/Save workspace as 的功能与 save 命令等价。需要注意的是，save 只能保存变量和数据，不能保存命令（保存命令行须通过程序设计 M 文件方式实现，见 §0.4）。

为了与其他应用程序交换数据，可能需要用 ASCII 码方式或格式化文件来传递数据。save 和 load 提供了写和读 ASCII 码数据文件的选项（详见 save 和 load 的帮助信息）。MATLAB 还允许使用 C 语言读写命令 fprintf，fscanf，fopen，fread 等来传递格式化数据文件，其使用格式与 C 语言基本一致。

§0.2 矩阵和数组运算

MATLAB 基本数据单元是无需指定维数的矩阵，标量可看作 1×1 矩阵，n 维行向量或列向量分别可看作 $1\times n$ 或 $n\times1$ 矩阵。

1. 矩阵的输入

输入矩阵最基本的方法是直接输入矩阵的元素，用中括号[ ]表示矩阵，同行元素间用空格或逗号分隔，不同行间用分号或回车分隔，例如

```
》clear; a=[1, 2, 3; 4, 5, 6; 7, 8, 9]
a=
    1     2     3
    4     5     6
    7     8     9
```

或

```
》a=[1  2  3                      %这种方式特别适用于大型矩阵
4 5 6
7 8 0]
a=
```

```
    1     2     3
    4     5     6
    7     8     0
```

对于等差数列构造的向量,可用冒号运算生成

```
》b=0:pi/10:pi              %初值:增量:终值
b=
Columns 1 through 7
    0    0.3142    0.6283    0.9425    1.2566   1.5708    1.8850
Columns 8 through 11
  2.1991    2.5133    2.8274    3.1416
》b2=1:10            %增量缺省值为 1
h2=
  1    2    3    4    5    6    7    8    9    10
》size (a), length (b)  %矩阵的阶数,向量的长度
ans=
    3    3
ans=
    11
```

矩阵元素双下标编址按通常方式,单下标编址按列排序,一律从 1 开始编址(注意下标不能用 0)。利用下标可获得矩阵元素或修改矩阵,例如

```
》a(3, 2), a(6)
ans=
    8
ans=
    8
》a(3, 4)=1                 %将第 3 行第 4 列元素设为 1
a=
    1    2    3    0
    4    5    6    0
    7    8    0    1
```

可以从大矩阵中抽取一个小矩阵,

```
》c=b(1:4)
c=
    0    0.3142    0.6283    0.9425
》d=a([1  3], :)             %取 a 的第 1, 3 行,所有列
d=
    1    2    3    0
    7    8    0    1
```

小矩阵也可拼成大矩阵,

```
» e=[c; d]
e=
          0    0.3142    0.6283    0.9425
     1.0000    2.0000    3.0000         0
     7.0000    8.0000         0    1.0000
```

一些特殊矩阵可用内部函数产生，如 ones (m, n)生成元素全为 1 的 m 行 n 列矩阵，zeros (m, n)生成元素全为 0 的 m 行 n 列矩阵等。

矩阵的输入还有其他方式，如用文件产生，用剪贴板写入，从外部数据文件读入等。

2. 矩阵运算符

矩阵有多种运算。矩阵运算符见表 0-2。

表 0-2 矩阵运算符

运　算	符　号	说　明
转置	A'	复矩阵共轭转置，实转置用 A.'
加与减	A+B 与 A-B	
数乘矩阵	k*A 或 A*k	
矩阵乘方	A^k	
数与矩阵加减	k+A 与 k-A	k+A 等价于 k*ones(size (A))+A
矩阵除法	左除 A\B，右除 B/A	它们分别为 AX＝B 和 XA＝B 的解

```
» clear;
» A=[1  -1; 0  2]; B=[0  1; 1  -1];
» A*B
ans=
    -1     2
     2    -2
» A\B                %即 A^-1B
ans=
    0.5000    0.5000
    0.5000   -0.5000
» A/B                %即 AB^-1
ans=
    0    1
    2    0
» A+100
ans=
    101    99
    100   102
```

3. 数组运算符(点运算符)

点运算符.*、./、.\、.^表示矩阵对应元素的运算，在作图、编写函数时经常使用。

```
》A.*B                     %注意与 A*B 的区别
ans=
     0    -1
     0    -2
》A.\B
ans=
          0   -1.0000
        Inf   -0.5000
》A.^2
ans=
     1     1
     0     4
》1./A                     %等价于 ones(size(A))./A
ans=
     1.0000   -1.0000
        Inf    0.5000
```

4. 关系与逻辑运算

MATLAB 有 6 个关系运算和 3 个逻辑运算符，都是对于元素的操作：

<	小于；	~=	不等于；
<=	小于等于；	&	与；
>	大于；	|	或；
>=	大于等于；	~	非。
==	等于；		

在 MATLAB 中，“真”用 1 表示，“假”用 0 表示，而逻辑运算中，所有非零元素作为 1 处理。

```
》A>B
ans=
     1     0                %想一想，什么意思？
     0     1
》A&B
ans=
     0     1
     0     1
```

5. 数学函数

矩阵的数学函数也是按元素的运算，使用通常的函数号，如 sin(A)，asin(A)，tan(A)，exp(A)，sqrt(A)，log(A)等，例如

```
》sqrt(A)
ans=
   1.0000             0+1.0000i
        0        1.4142
```

§0.3 图　形

plot	基本二维图形；	clabel	等高线高度标志；
fplot	一元函数图象；	text	文本；
plot3	空间曲线；	grid	格栅；
meshgrid	网格数据生成；	legend	图例；
mesh	网面图；	hold	图形保持；
surf	曲面图；	axis	定制坐标轴；
contour	等高线图；	view	改变视点；
contour3	3 维等高线图；	subplot	子图；
title	标题；	figure	新图形窗口；
xlabel	x 轴说明；	clf	清除图形；
ylabel	y 轴说明；	close	关闭图形窗口。
zlabel	z 轴说明；		

1. 曲线图

plot (x, y)　　作出以数据($x(i)$, $y(i)$)为节点的折线图，其中 x, y 为同长度的向量；

plot (x1, y1, x2, y2, ...)　　作出多组数据折线图；

fplot ('fun', [a, b])　　作出函数 fun 在区间[a, b]上的函数图；

plot3 (x, y, z)　　空间曲线图，其中 x, y, z 为同长度的向量。

» plot ([1　4　2　5], [3　5　1　4])　　%依次连接(1, 3), (4, 5), (2, 1), (5, 4)四点(如图 0-2)

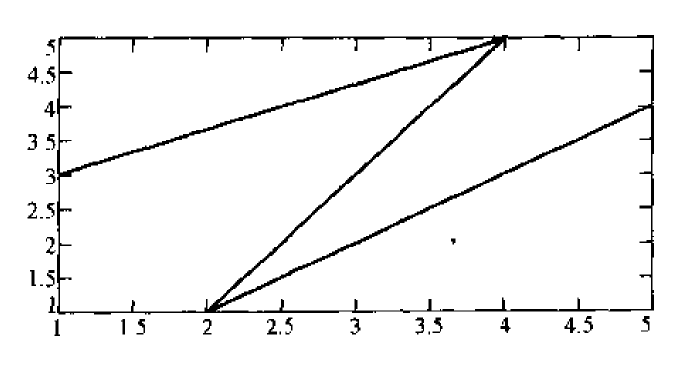


图 0-2　plot 的使用

» plot ([1　4　2　5], [3　5　1　4], 1:3, [2　10　3])　　%注意颜色

图形可直接保存为 M 文件，也可利用图形窗口菜单 Edit/Copy figure 将其作为图片剪贴到 Word 或其他应用程序中。

图形的线型，标记，颜色均可设定。常用的见表 0-3。

表 0-3　图形元素的设定

颜色		线型		标记	
b	蓝(默认)	-	实线(默认)	无标记(默认)	
g	绿	--	划线	*	星
r	红	:	虚线	.	点
y	黄	-.	点划线	o	圈
m	洋红			x	叉
c	青			+	十字
w	白			s	方块
k	黑			d	菱形
				V	下三角形
				Λ	上三角形
				<	左三角形
				>	右三角形
				h	六角星
				p	五角星

例 1　一元函数图 $y=x^3-x-1$ 和 $y=|x|^{0.2}\sin(5x)$，$-1<x<2$（结果见图 0-3）

```
» fplot ('x.^3-x-1', [-1, 2]); hold on;       % hold on 在作下一幅图时保留已有图象
» x=-1:0.2:2; y=abs(x).^2.*sin(5*x);
» plot (x, y, ':ro'); hold off;               % hold off 释放 hold on
```

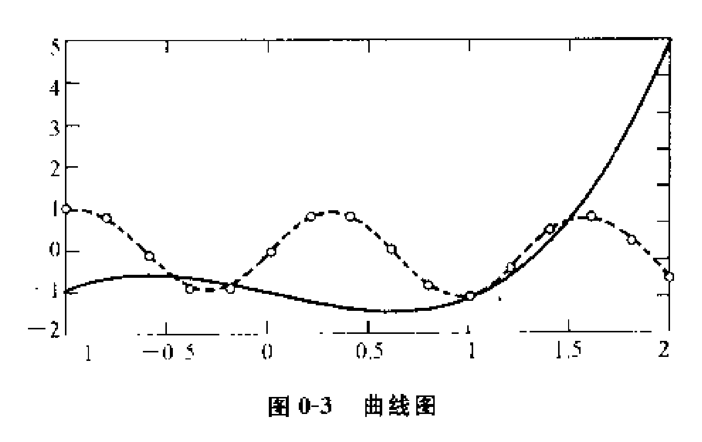


图 0-3　曲线图

2. 曲面图

[x, y] = meshgrid (xa, ya)　当 xa, ya 分别为 m 维和 n 维行向量，得到 x 和 y 均为 n 行 m 列矩阵。meshgrid 常用于生成 x-y 平面上的网格数据；

mesh (x, y, z)　绘制网面图，是最基本的曲面图形命令，其中 x、y、z 是同阶矩阵，表示曲面三维数据；

mesh (xa, ya, z)　xa, ya 分别为 m 维和 n 维向量，z 为 n 行 m 列矩阵，等价于先[x, y] = meshgrid (xa, ya)再 mesh (x, y, z)；

surf (x, y, z)　绘制曲面图,与 mesh 用法类似;
contour (x, y, z)　绘制等高线图,与 mesh 用法类似,可指定 z 的范围;
contour3 (x, y, z)　绘制三维等高线图,与 mesh 用法类似,可指定 z 的范围。

例如

```
» xa = 1:3, ya = 1:4; [x, y] = meshgrid (xa, ya); z = x + y; mesh (x, y, z)
» [x, y, z]
```

1	2	3	1	1	1	2	3	4
1	2	3	2	2	2	3	4	5
1	2	3	3	3	3	4	5	6
1	2	3	4	4	4	5	6	7

这 3 组数据构成网面的 12 的格点坐标。

例 2　二元函数图 $z = x\exp(-x^2 - y^2)$

```
» clear; close;                  %close 关闭已有图形窗口
» xa = - 2:.2:2; ya = xa;
» [x, y] = meshgrid (xa, ya); z = x.* exp( - x.^2 - y.^2);
» mesh (x, y, z);                %网面图,看看图(见图 0-4)
» surf (x, y, z);                %曲面图,看看图
» contour (x, y, z);             %等高线图,看看图
» contour3 (x, y, z);            %三维等高线图,看看图
```

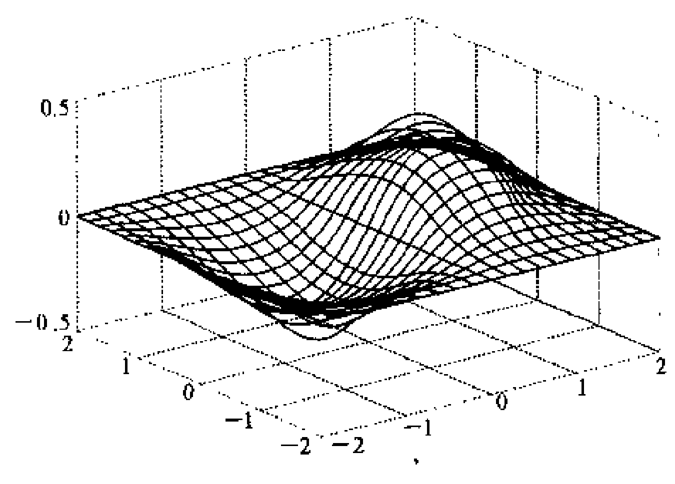


图 0-4　曲图图

3. 说明、格栅和图例

title ('字符串')　图形标题说明;
xlabel, ylabel, zlabel　用法类似于 title,分别说明坐标轴 x, y, z;
text (x, y, '字符串')　将字符串表达的文字标于(x, y)处;
legend ('字符串 1', '字符串 2', ...)　依次说明图例;
grid on/grid off　显示/不显示格栅。

例 3　参变量函数

$$\begin{cases} x = e^{-0.2t}\cos\dfrac{\pi}{2}t \\ y = e^{-0.2t}\sin\dfrac{\pi}{2}t \qquad (0 < t < 20) \\ z = t \end{cases}$$

打印图象如图 0-5 所示。

```
» clear; close;
» t = 0:0.1:20; r = exp(-0.2*t); th = 0.5*pi*t;
» x = r.*cos(th); y = r.*sin(th); z = t;
» plot3 (x, y, z);
» title ('SPACE LINE');
» text (x(1), y(1), z(1), 'START')
» n = length(x); text(x(n), y(n), z(n), 'END')
» xlabel('X'); ylabel('Y'); zlabel('Z');
» legend ('Cone Line');      %图例可以拖动
» grid on;                   %再试一试 grid off，观察变化
```

图 0-5　空间曲线图

4. 图形控制

hold on/hold off　保留/释放现有图形；
axis off/on　不显示/显示坐标轴；
axis ([a, b, c, d])　定制 2 维坐标轴范围 $a < x < b$, $c < y < d$;
axis ([a, b, c, d, e, f])　定制 3 维坐标轴范围 $a < x < b$, $c < y < d$, $e < z < f$;
view (az, el)　调整视角，az 为角度，el 为高度（默认 az = -37.5, el = 30）；
figure　开一个新图形窗口；
close　关闭现有图形窗口；
subplot (m, n, k)　将图形窗口分为 mn 个子图，并指向第 k 幅图。

5. 图形对象设置

MATLAB 图形窗口、坐标轴、图形元素等往往是系统自动选定的，有些可用图形命令来改变，如曲线的颜色、标题文本、视点等。一般来说可使用对象设置命令 set 来设置，如文本的字体、坐标轴刻度、窗口的背景色等，但用法较复杂。MATLAB5.3 的图形窗口建立了图形对象编辑工具，可直接改变图形设置（图 0-6）。

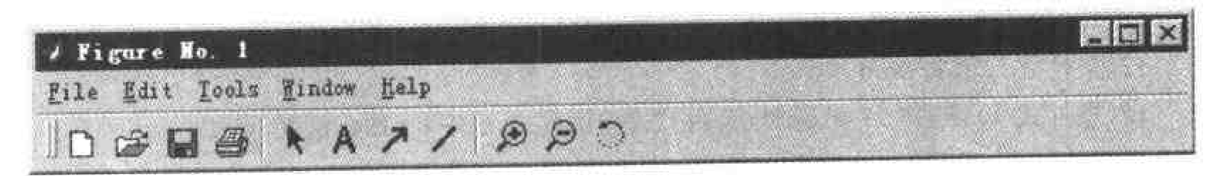


图 0-6　图形窗口菜单和工具栏

作为示例，我们现在来设法将例 3 产生的图形改变成图 0-7 样子。

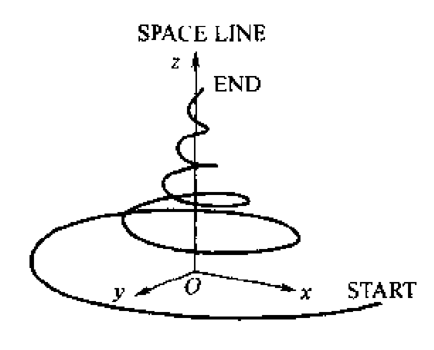


图 0-7 图形对象设置

(1) 按工具栏 Enable 按钮(指向左上方的箭头),使图形处于编辑状态;

(2) 曲线粗细和颜色:选中曲线,用鼠标双击可编辑,将颜色改为红色,粗细改为 3;

(3) 字体大小:选中所要编辑的文字,用菜单 Tools/Text property…设置;

(4) 坐标轴:选中并双击坐标轴,将 y 轴方向设为"reverse";

(5) 再按工具栏 Enable 按钮,使图形回到正常状态;

(6) 调整视角:按工具栏 Rotate 按钮(带箭头的圆圈),选中坐标轴,调整为理想角度;

(7) 加箭头线:按工具栏 Add arrow 按钮(指向右上方的箭头),从原点拉出三根坐标线,并双击编辑加粗;

(8) 加文本:按工具栏 Add text 按钮(字母 A),在适当位置写 X, Y, Z, O,设置字体和大小;

(9) 在命令窗口用命令 axis off 使原坐标轴不显示。

但图形窗口对象编辑工具并没有给出所有设置,更详细的设置可用菜单 File/Property Editer ... 完成(MATLAB5.2 这一菜单在命令窗口)。

§0.4 程 序 设 计

1. 控制流

到目前为止,我们用的命令都是顺序结构的,对于复杂的计算,需要循环和分支结构。

例 4 计算 $s=\sum_{n=1}^{100}\frac{1}{n^2}$

```
》clear; s=0;
》for n=1:100
  s=s+1/n/n;
end
》s
s=
  1.6350
```

常用控制流语句有

```
for 循环变量=初值:增量:终值,语句;end
while(条件式),语句;end
if(条件式),语句;end
if(条件式1),语句1; elseif(条件式2),语句2;……; else,语句;end
switch(分支变量)case(值1),语句1; case(值2),语句2;……;
                    otherwise 语句;end
```

另外还有 break(中断循环),return(中断执行返回)等。

2. M脚本文件

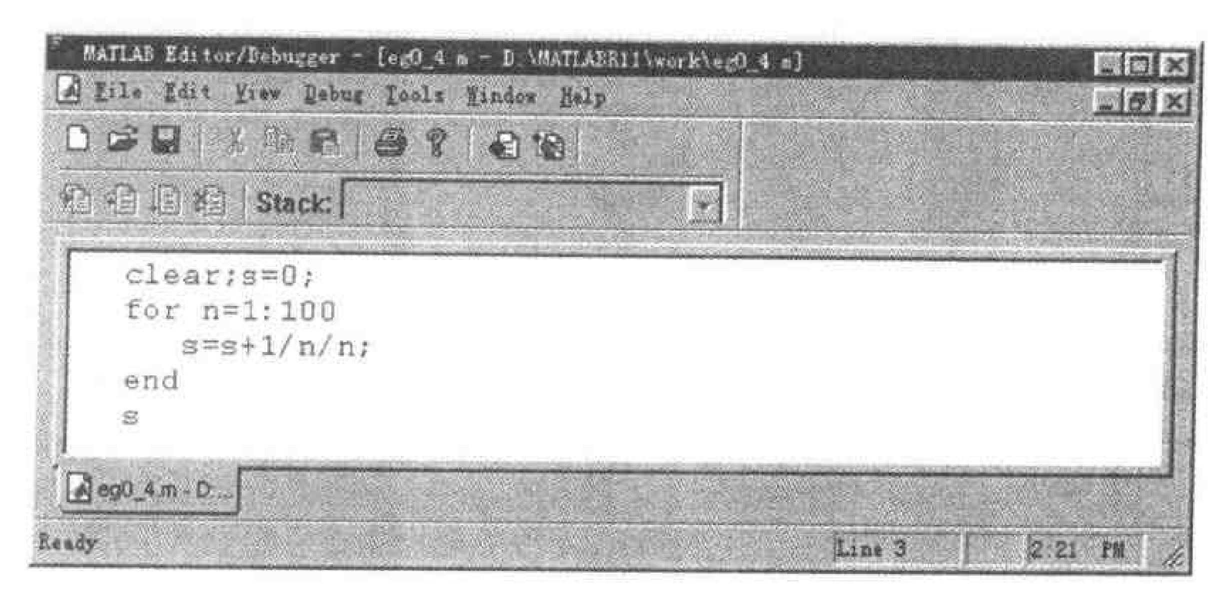


图 0-8 M脚本(函数)文件编辑器

从工具栏的编写程序(New File)按钮或在 File/New 菜单选 M-file 就进入 MATLAB 的编辑器 Editer,用以编写用户的 M 文件。如图 0-8。M 文件可分为两类:M 脚本文件和 M 函数文件。将多条 MATLAB 语句写在编辑器中保存在适当的目录中(这个目录须在 MATLAB 搜索路径中),就得到一个 M 脚本。如我们将例 4 中的几条语句写在编辑器中,

```
clear; s = 0;
for n = 1:100
  s = s + 1/n/n;
end
s
```

保存为 eg0_4(不要加扩展名 m),然后在命令窗口执行

```
» eg0_4
s=
  1.6350
```

使用编辑器也可打开和修改 M 文件,观察变量值,调试程序等。但要注意:每次修改程序后都要存盘。文件名以字母开头,由字母、数字或下划线组成,一般取 3 个字符以上。注意不要含非法字符如空格、减号等。

3. 函数文件

M 函数文件以 function 开头,格式为

```
function  输出变量 = 函数名(输入变量)
语句;
```

在 MATLAB 中,使用 M 函数是以该函数的磁盘文件主名调用,而不是 function 中的函数名,但我们建议二者用同名。M 函数与 M 脚本有两个重要区别:第一,M 函数一般有输入

输出变量；第二，M 函数中所有变量为局部变量，而脚本文件中所有变量都存在于命令窗口，即全局变量。例如，在编辑器窗口写函数文件

```
%M 函数 eg0_4f.m
function s = f(m)
s = 0;
for n = 1:m
    s = s + 1/n/n;
end
```

保存为 eg0_4f.m，在命令窗口执行

```
» clear; eg0_4f(100), eg0_4f(1000)
ans=
      1.6350
ans=
      1.6439
» s
??? Undefined function or variable s.        %说明 s 不是命令窗口的变量
```

4. *编程几点高级事项

input	提示输入；	eval	执行命令；
disp	强行显示；	feval	函数求值；
num2str	数值转字符串；	keyboard	键盘指令；
str2num	字符串转数值；	pause	延时；
int2str	整数转字符串；	warning	显示警告；
mat2str	矩阵转字符串；	error	显示错误并中断；
char	按 ASCII 码转换；	nargin	函数的输入变量个数；
strcmp	字符串比较；	nargout	输出变量个数。

(1) 注释：注释语句用%开头，对本行后面字符起作用，注释语句不参与运算，起说明作用，增强程序的可读性，一个好的 M 文件开头应有一段注释，说明功能和使用方法，这部分注释使用

help <u>文件主名</u>

可看到。注释符%也常用于程序调试。

(2) 提示输入：input 和 disp 是一组输入输出语句，例如编写下列脚本文件 finddet.m，使对键盘输入的方阵求得行列式。

```
%文件 finddet.m
clear A;
A = input ('Enter a square matrix:');
d = det(A);
disp(['Its determinant is ', num2str(d)]);
```

这里 num2str 是将数转换为字符串，这样可与前面的字符串一起构成字符串矩阵。现在运行

```
» finddet
Enter a square matrix:(请你输入方阵,如[1 2; 3 4])
Its determinant is -2
```

(3) 子函数:M 函数中允许使用子函数。M 函数中第一个 function 为主函数,其他 function 为子函数。

(4) 全局变量:M 文件间变量值传递除使用参量外还有一种方式:定义 global(全局变量),它的意义与通常全局变量稍有区别,只对有定义的文件起作用。

(5) 数组化编程:MATLAB 是数组化程序语言,好的 M 文件应尽量使用内部函数(包括关系与逻辑函数 any、all、find 等,见例 5),少用循环语句,以提高运算速度。如例 4 写成下列形式运算较快。

```
%M 函数 eg0_4f.m 的改进
function s = f(m)
n = 1:m;
s = sum(1./n.^2);    % sum 表示向量的和
```

(6) 预分配:尽管 MATLAB 数组无须定义大小,但经常改变数组大小会影响速度,采取一些预分配方法可提高运算速度。如

较差的程序	较好的程序
v(1)=2;	v=2*ones(1, 100);
for i=2:100	for i=2:100
v(i)=v(i-1)^0.5;	v(i)=v(i-1)^0.5;
end	end

(7) 与 C/C++及 FORTRAN 的接口:MATLAB 提供了 API 函数将 C/C++或 FORTRAN 程序转化为在 MATLAB 中运行的 MEX 文件,也可用 Complier 从 M 函数生成 C/C++代码或可执行文件。MATLAB 还可与 EXCEL 交换数据。

*例 5 分段函数图

$$p(x, y)=\begin{cases}0.5457\exp(-0.75y^2-3.75x^2-1.5x), & x+y>1\\ 0.7575\exp(-y^2-6x^2), & -1<x+y\leqslant 1\\ 0.5457\exp(-0.75y^2-3.75x^2+1.5x), & x+y\leqslant -1\end{cases}$$

```
%M 文件 eg0_5.m
clear; close;
xa = -2:0.1:2; ya = -2:0.1:2; [x, y] = meshgrid (xa, ya);
z = zeros(size(x));     % 预处理可加快速度
k1 = find (x+y>1);          % k1 返回符合 x+y>1 的数组编址(单下标)
z(k1) = 0.5457*exp(-0.75*y(k1).^2-3.75*x(k1).^2-1.5*x(k1));
k2 = find(x+y<=1&x+y>-1);
z(k2) = 0.7575*exp(-y(k2).^2-6*x(k2).^2);
k3 = find(x+y<-1);
z(k3) = 0.5457*exp(-0.75*y(k3).^2-3.75*x(k3).^2+1.5*x(k3));
```

```
mesh(x, y, z);
```

§0.5 在线帮助和文件管理

1. 在线帮助

help:	显示查询目录;
help 子目录名	显示子目录中所有 MATLAB 系统命令及函数;
help 命令或函数	显示该命令或函数的说明部分;
lookfor 关键词	显示与该关键词有关的命令和函数。

```
» help inv
INV    Matrix inverse.
    INV(X) is the inverse of the square matrix X.
    A warning message is printed if X is badly scaled or
    nearly singular.
» lookfor inverse
ACOS   Inverse cosine.
ACOSH  Inverse hyperbolic cosine.
ACOT   Inverse cotangent.
  ... ... ... ... ... ...
INV   Matrix inverse.
PINV   Pseudoinverse.
IFFT   Inverse discrete Fourier transform.
  ... ... ... ... ... ...
```

2. 文件管理

MATLAB 文件有 M、Mat、Mex 等。其中 M 文件是最主要的,MATLAB 绝大多数内部函数是 M 文件,我们用户自编的程序一般也是 M 文件。

what	显示当前目录中的 MATLAB 文件;
dir	显示当前目录中所有文件;
type M 文件主名	显示指定的 M 文件内容;
which M 文件主名	显示指定的 MATLAB 文件的路径;
cd	显示当前工作目录;
cd 子目录名	进入子目录;
mkdir	建子目录;
! DOS 命令	执行 DOS 命令。

例如

```
》which ifft
D:\matlab11\toolbox\matlab\datafun\ifft.m
》type ifft
```
显示 M 文件 ifft.m 的全部内容。

```
》type acos
acos is a built-in function.          %一些核心函数不是 M 文件
```

3. MATLAB 工作目录

MATLAB 只执行当前目录和搜索路径(写在文件 pathdef.m 中)中的命令和函数。当 MATLAB 接受到一个命令首先检查是否为命令窗口常量或变量,然后检查当前工作目录 M 文件名,再依次按路径队列搜索,排在后面的同名 M 文件得不到执行。你所正在执行的 M 文件的位置可用 which 查到,并可用 type 显示文件内容。

使用工具栏里的路径浏览器(Path Browser)可查看到搜索路径队列,也可改变当前工作目录或在搜索路径中添加新目录。对于使用公共计算机的读者,我们建议设置你自己的工作目录(如软盘 a:)。每次进入 MATLAB,使用路径浏览器将你的目录设置为当前目录,你编写的程序都保存于此目录,就不会与别人的程序冲突了。

在 MATLAB 中,当你对某命令或函数的使用不清楚时,多使用 help,多用例子试一试;想阅读程序,使用 type;若你想用关键字查询命令和函数,使用 lookfor。这样你不用书面材料,就可自学 MATLAB 了。

§0.6* 符号数学工具箱

符号数学工具箱(Extended Symbolic Math)是 MATLAB 一个特殊的工具箱,可进行解析数学运算和任意指定精度数值计算。

1.* 符号对象

符号运算使用一种特殊的数据类型,称为符号对象(Symbolic Object),用字符串形式表示,但又不同于普通字符串(Char Array)。其变量、表达式均为符号对象。符号对象使用 sym 或 syms 生成,如

```
》syms a b c;   %生成几个符号变量
》A=[0.1 a; b c]   %当然达是一个符号矩阵,其输出表达与数值矩阵有明显区别
A=
[1/10,  a]
[  b,   c]
》d=sym('5^0.5')   %数值表达式转化为符号表达式
d=
  5^0.5          %注意它已不是数值
》f=sym('x^3-1')   %字符串化为符号对象
f=
  x^3-1                 %注意它已不是字符串
》syms x; f=sym(x^3-1)   %结果同上
```

```
» str = char(f)       %符号对象转化为字符串
» n = double(d)       %符号对象转化为数值
```

现在请使用工具栏的工作空间浏览器(Workspace Browser),观察各变量数据类型。可见,符号对象占用字节数远大于数值或字符,同时其运算速度也慢许多,所以通常 Symbolic 命令只作为"符号计算器"作解析运算,数值计算一般不提倡用 Symbolic 命令。

2. * 基础符号运算函数

sym	字符串或数值到符号的转换;	expand	展开;
char	符号到字符串的转换;	collect	合并同类项;
double	符号到数值的转换;	simplify, simple	化简;
digits	计算字长设置;	subs	变量代换;
vpa	任意精度计算;	ezplot	符号函数作图;
factor	因式分解;	funtool	函数计算器。

注:(1)MATLAB 大多运算符如+, -, *, /, ^, .*, ./, .^等适用于符号对象;(2)funtool 是一个直观的图形化函数计算器,很方便进行函数代数运算和微积分。自己试一试。

*例 6 (多项式运算) $f(x)=(x-1)^3$, $g(x)=(x+1)^3$

```
» syms x; f = (x-1)^3; g = (x+1)^3;
» h = f*g
h=
(x-1)^3*(x+1)^3
» s = expand(h)
  s=
  x^6-3*x^4+3*x^2-1
» hf = factor(s)
hf=
(x-1)^3*(x+1)^3
» hsub = subs (s, x, x^2+x+1)              %用 x^2+x+1 替换 s 中的 x
hsub=
(x^2+x+1)^6-3*(x^2+x+1)^4+3*(x^2+x+1)^2-1
» hsim = simple(hsub)
  hsim=
  x^3*(x+1)^3*(x^2+x+2)^3
» hnum = subs (hsim, pi)             %x=pi 时的值
hnum=
  7.4507e+006
» hnum2=subs (hsim, 'pi')           %注意不是 pi
hnum2=
pi^3*(pi+1)^3*(pi^2+pi+2)^3
» vpa (hnum2, 50)              %50 位精度计算
ans=
```

7450711.392792243259113843088323110810285249299621 3

§0.7 习　　题

1　实践下列 MATLAB 命令，观察结果，以理解 ；，… 的作用。

(i) r = 2; v = 4/3* pi* r ^ 3

(ii) r = 2, v = 4/3* pi* r ^ 3

(iii) r = 2; v = 4/3* ...

　　pi* r ^ 3

2　观察下列计算结果，理解数组与矩阵运算的意义

(i) [1　2; 3　4] + 1

(ii) [1　2; 3　4]* [1.1　1.2; 1.3　1.4]

(iii) [1　2; 3　4].* [1.1　1.2; 1.3　1.4]

(iv) sin (1:4)

(v) [1　2; 3　4]> = [4, 3; 2　1]

3　作出下列函数图象

(i) $y = x^2 \sin(x^2 - x - 2)$　　$-2 \leqslant x \leqslant 2$　　(分别使用 plot 或 fplot 完成)

(ii) $x^2/4 + y^2/9 = 1$ (椭圆　提示：使用参数方程)

(iii) $z = x^2 + y^2$ (抛物面)　　$|x| < 3, |y| < 3$

4　用循环语句形成 Fibonacci 数列 $F_1 = F_2 = 1, F_k = F_{k-1} + F_{k-2}, k = 3, 4, \cdots$。并验证极限

$$\frac{F_k}{F_{k-1}} \to \frac{1+\sqrt{5}}{2}$$

(提示：计算至两边误差小于精度 1e−8 为止)

实验一　投入产出平衡　矩阵和线性方程组

本实验中我们学习 MATLAB 有关线性代数运算的命令，加深对向量和矩阵概念和理论的理解，并研究投入产出分析、基因遗传等应用问题。补充知识介绍了数值计算中常常涉及的范数、条件数和病态方程组等概念以及线性方程组 Gauss 消去法。

§1.1　引例：国民经济投入产出综合平衡

在一个国家或区域的经济系统中，各部门（或企业）既有投入又有产出。生产的产品满足系统内部各部门和系统外的需求，同时也消耗系统内各部门提供的产品。美国哈佛大学教授 Leontief 于 20 世纪 30 年代首先提出并成功建立了国民经济的投入产出的数学模型，并主持制定了美国国民经济投入产出综合平衡计划。因此他获 1973 年诺贝尔经济学奖。

设有 n 个经济部门，x_i 为部门 i 的总产出，c_{ij} 为部门 j 单位产品对部门 i 产品的消耗，d_i 为外部对部门 i 的需求，f_j 为部门 j 新创造的价值。那么各经济部门总产出应满足下列关系式

$$x_i = \sum_{j=1}^{n} c_{ij} x_j + d_i, \quad i = 1, \cdots, n \tag{1.1}$$

$$x_j = x_j \sum_{i=1}^{n} c_{ij} + f_j, \quad j = 1, \cdots, n \tag{1.2}$$

上述(1.1)式称为**分配平衡方程组**，(1.2)式称为**消耗平衡方程组**。由于各 x_i 相互关联，不可能独立地进行研究。我们可以用矩阵理论来表达和分析。令 $C = (c_{ij})$，$\boldsymbol{X} = (x_1, \cdots, x_n)'$，$\boldsymbol{D} = (d_1, \cdots, d_n)'$，$\boldsymbol{F} = (f_1, \cdots, f_n)'$，将(1.1)式化为矩阵形式

$$\boldsymbol{X} = C\boldsymbol{X} + \boldsymbol{D} \tag{1.3}$$

令 $A = E - C$，这里 E 为单位矩阵。那么(1.3)式化为

$$A\boldsymbol{X} = \boldsymbol{D} \tag{1.4}$$

经济学上 C 称为**直接消耗矩阵**，A 称为 **Leontief 矩阵**。令

$$B = C\begin{pmatrix} x_1 & & & \\ & x_2 & & \\ & & \ddots & \\ & & & x_n \end{pmatrix}, \quad \boldsymbol{Y} = [1, 1, \cdots, 1]B \tag{1.5}$$

B 表示各部门间的投入产出关系，称为**投入产出矩阵**，Y 表示各部门的总投入，称为**总投入向量**。那么新**创造价值向量**

$$\boldsymbol{F} = \boldsymbol{X} - \boldsymbol{Y}' \tag{1.6}$$

§1.2 数学理论复习：数性代数

自然科学和工程实践很多问题的解决都归纳为线性代数方程组的求解和矩阵运算。有些问题本身就是一个线性方程组，例如结构应力分析问题、电子传输网分析问题、投入产出分析问题和各种晶体管电路分析问题；另一方面有些数值计算方法导致线性方程组求解，如数据拟合问题、非线性方程组和偏微分方程组数值解问题等等。

1. 线性方程组

n 个未知量 m 个方程的线性方程组一般形式为

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \quad\cdots \quad \cdots \quad \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases} \tag{1.7}$$

令

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \ \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \ \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

则得矩阵形式

$$A\boldsymbol{x} = \boldsymbol{b} \tag{1.8}$$

若右端 $\boldsymbol{b} = \boldsymbol{0}$，即

$$A\boldsymbol{x} = \boldsymbol{0} \tag{1.9}$$

则称方程组为**齐次的**。

方程组(1.8)可能有唯一解，可能有无穷多解，也可能无解，主要取决于系数矩阵 A 及增广矩阵$(A, \boldsymbol{b})$的秩。若秩(A) = 秩$(A, \boldsymbol{b}) = n$，存在唯一解，其解理论上可用 Cramer 法则求出，但由于这种方法要计算 $n+1$ 个 n 阶行列式，计算量太大通常并不采用；若秩(A) = 秩$(A, \boldsymbol{b}) < n$，存在无穷多解，其通解可表示为对应齐次方程组(1.9)的一个基础解系与(1.8)式的一个特解的叠加；若秩$(A) \neq$ 秩$(A, \boldsymbol{b})$，则无解，这时一般寻求最小二乘近似解，即求 $\boldsymbol{x}$ 使向量 $A\boldsymbol{x} - \boldsymbol{b}$ 长度最小。

2. 逆矩阵

方阵 A 称为**可逆的**，如果存在方阵 B，使

$$AB = BA = E$$

这里 E 表示单位阵。并称 B 为 A 的**逆矩阵**，记 $B = A^{-1}$。方阵 A 可逆的充分必要条件是 $|A| \neq 0$。求逆矩阵理论上的公式为

$$A^{-1} = \frac{1}{|A|}A^* \tag{1.10}$$

这里 A^* 为 A 的伴随矩阵。利用逆矩阵,(1.2)式的解可表示为

$$x = A^{-1}b$$

由于这个公式涉及大量行列式计算,数值计算不采用。求逆矩阵的数值算法一般是基于求解线性方程组的方法。

3. 特征值与特征向量

对于方阵 A,若存在数 λ 和非零向量 x 使

$$Ax = \lambda x \tag{1.11}$$

则称 λ 为 A 一个**特征值**,x 为 A 的一个对应于特征值 λ 的**特征向量**。特征值计算归结为特征多项式的求根。对于 n 阶实数方阵,特征多项式在复数范围内总有 n 个根。对应于特征值 λ 的特征向量是齐次线性方程组

$$(A - \lambda E)x = \mathbf{0} \tag{1.12}$$

的所有非零解。通常只需要求它的一组线性无关解。特征值和特征向量求解的数值方法是相当复杂的,适用性较广的是正交三角分解系列算法(参见文献[9])。

§1.3 线性代线运算 MATLAB 命令

MATLAB 是矩阵化程序设计语言,所以处理矩阵和向量运算特别方便。关于矩阵和向量的一些基本运算命令已在预备知识中介绍,常用命令和函数还有

zeros	生成元素全为 0 的矩阵;	eig	特征值与特征向量;
ones	生成元素全为 1 的矩阵;	diag	对角阵;
eye	生成单位矩阵;	trace	方阵的迹;
linspace	生成等距行向量;	rank	矩阵的秩;
rand	生成随机矩阵;	rref	矩阵的行最简形;
det	方阵的行列式;	orth	正交规范化;
inv	方阵的逆;	null	求基础解系;
norm	矩阵或向量范数;	jordan	Jordan 标准形分解。
cond	方阵的条件数;		

1. 特殊矩阵生成

zeros(m, n)　生成 m 行 n 列的零矩阵;
ones(m, n)　生成 m 行 n 列的元素全为 1 的矩阵;
rand(m, n)　生成 m 行 n 列[0, 1]上均匀分布随机数矩阵;
eye(n)　生成 n 阶单位矩阵;
当 **A** 是矩阵,diag(A)返回 A 的对角线元素构成的向量;
当 **X** 是向量,diag(X)返回由 **X** 的元素构成的对角矩阵;
linspace(x1, x2, n)生成 x_1 与 x_2 间的 n 维等距行向量,即将[x_1, x_2]n－1 等分。

```
》ones(3, 3)
ans=
     1     1     1
     1     1     1
     1     1     1
》eye(3)
ans=
     1     0     0
     0     1     0
     0     0     1
》linspace(1, 2, 4)
ans=
    1.0000    1.3333    1.6667    2.0000
》rand(2, 4)    %由于随机性,每次结果不同
ans=
    0.6154    0.9218    0.1763    0.9355
    0.7919    0.7382    0.4057    0.9169
```

2. 行列式和逆矩阵

det (A)	返回方阵 A 的行列式;
inv (A)	返回 A 的逆矩阵。

例如

```
》A=[1  2; 3  4]; det(A), inv(A)
ans=
    -2
ans=
   -2.0000      1.0000
    1.5000     -0.5000
```

3. 矩阵除法

左除法 A\B	求解矩阵方程 AX = B;
右除法 B/A	求解矩阵方程 XA = B。

矩阵除法会根据 A 的特点自动选定合适的算法求解,然后尽可能给出一个有意义的结果。

(1) 当 A 为方阵,其结果与 inv(A)*B 基本一致;

(2) 当 A 不是方阵,除法将自动检测。若为超定方程组(即无解),除法将给出最小二乘意义上的近似解,即使得向量 $\boldsymbol{AX}-\boldsymbol{B}$ 的长度达到最小;若为不定方程组(即无穷多解),除法将给出一个具有最多零元素的特解;当然若为唯一解,除法将给出这个解。用户应对结果有一个正确认识,通常可使用 rank 比较系数矩阵的秩与增广矩阵的秩来分析。除超定方程

组外，除法对符号对象同样适用。

例1 解下列方程组

(1) 定解方程组 $\begin{cases} x+2y=1 \\ 3x-2y=4 \end{cases}$

(2) 不定方程组 $\begin{cases} x+2y+z=1 \\ 3x-2y+z=4 \end{cases}$

(3) 超定方程组 $\begin{cases} x+2y=1 \\ 3x-2y=4 \\ x-y=2 \end{cases}$

(4) 奇异方程组 $\begin{cases} x+2y=1 \\ -2x-4y=-2 \end{cases}$

```
》A=[1  2; 3  -2]; B=[1; 4]; x=A\B
x=                          %求得唯一解
     1.2500
    -0.1250
》A=[1  2  1; 3  -2  1]; B=[1; 4]; x=A\B
x=                          %求得一特解
     1.2500
    -0.1250
          0
》A=[1  2; 3  -2; 1  -1]; B=[1; 4; 2]; x=A\B
x=                          %求得一最小二乘近似解
     1.2838
    -0.1757
》A=[1  2; -2  -4]; B=[1; -2]; x=A\B
Warning: Matrix is singular to working precision.
x=                                    %不能直接求解
    Inf
    Inf
》A=[1  2; -2  -4; 0  0]; B=[1; -2; 0]; x=A\B  %增加0x+0y=0,使A不为方阵
Warning: Rank deficient, rank=1  tol=  2.9790e-015.
x=                                    %仍可求一特解
          0
     0.5000
```

例2 线性方程组的通解

$$\begin{cases} x_1-x_2+x_3-x_4=1 \\ -x_1+x_2+x_3-x_4=1 \\ 2x_1-2x_2-x_3+x_4=-1 \end{cases} \tag{1.13}$$

解：在无穷多解情况可用三种方法求得通解，第一种方法用 rref 化为行最简形以后求解；第二种方法用除法求出一个特解，再用 null 求得一个齐次组的基础解系。第三种方法是用符号数学工具箱中的 solve 求解(见实验二)。

```
》clear; a=[1  -1  1  -1; -1  1  1  -1; 2  -2  -1  1]; b=[1; 1; -1];
》[rank(a), rank([a, b])]
ans=
2  2                          % 秩相等且小于 4，说明有无穷多解
》rref([a, b])                % 方法一
ans=
     1   -1    0    0    0
     0    0    1   -1    1
     0    0    0    0    0         %通解 x1=x2, x3=x4+1(x2, x4 自由)
》x0=a\b, x=null(a)    % 方法二，注意 null 得到解空间的一个正交规范基
Warning: Rank deficient, rank=2  tol=  2.1756e-015.
x0=
   0
   0
   1
   0
x=
   -0.7071         0
   -0.7071   -0.0000
   -0.0000    0.7071
         0    0.7071          %通解为 k1*x(:, 1)+k2*x(:, 2)+x0
```

4. 特征值和特征向量

[V, D]=eig(A)返回方阵 A 的特征值和特征向量。其中 D 为特征值构成的对角阵，每个特征值对应的 V 的列为属于该特征值的一个特征向量，每个特征向量都是单位向量(即模等于 1)，并且属于同一特征值的线性无关特征向量已正交化。如果只有一个返回变量，则得到特征值构成的列向量。

```
》A=[1  2  2; 2  4  4; 2  4  4]; [V, D]=eig(A), t=eig(A)
V=
    0.9333    0.1333    0.3333
   -0.1333   -0.7333    0.6667
   -0.3333    0.6667    0.6667
D=
    0.0000         0         0
         0   -0.0000         0
```

```
       0          0      9.0000
t=
    0.0000
   -0.0000
    9.0000
```

§1.4 实验例题

现在我们来看一个小型的投入产出问题。

例 3 (投入产出分析)某地区有三个重要产业,一个煤矿,一个发电厂和一条地方铁路。开采一元钱的煤,煤矿要支付 0.25 元的电费及 0.25 元的运输费;生产一元钱的电力,发电厂要支付 0.65 元的煤费,0.05 元的电费及 0.05 元的运输费;创收一元钱的运输费,铁路要支付 0.55 元的煤费和 0.10 元的电费,在某一周内煤矿接到外地金额 50000 元定货,发电厂接到外地金额 25000 元定货,外界对地方铁路没有需求,问三个企业间一周内总产值多少才能满足自身及外界需求?三个企业间相互支付多少金额?三个企业各创造多少新价值?

解:这是一个投入产出分析问题。设 x_1 为本周内煤矿总产值,x_2 为电厂总产值,x_3 为铁路总产值,则

$$\begin{cases} x_1 - (0 \times x_1 + 0.65x_2 + 0.55x_3) = 50000 \\ x_2 - (0.25x_1 + 0.05x_2 + 0.10x_3) = 25000 \\ x_3 - (0.25x_1 + 0.05x_2 + 0 \times x_3) = 0 \end{cases}$$

产出向量 $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$,外界需求向量 $D = \begin{bmatrix} 50000 \\ 25000 \\ 0 \end{bmatrix}$,直接消耗矩阵 $C = \begin{bmatrix} 0 & 0.65 & 0.55 \\ 0.25 & 0.05 & 0.10 \\ 0.25 & 0.05 & 0 \end{bmatrix}$,Leontief 矩阵 $A = E - C$。根据(1.11)式、(1.12)式和(1.13)式求解如下。

```
» clear; C=[0  0.65  0.55; 0.25  0.05  0.1; 0.25  0.05  0];
» D=[50000; 25000; 0]; A=(eye(3)-C)
» X=A\D
»B=C*diag(X)
»Y=ones(1,3)*B
»F=X-Y'
```

得投入产出分析表,见表 1-1。

例 4 (隐性病遗传)在常染色体遗传中,后代是从父母的基因对中各继承一个基因,形成自己的基因型。如果我们所考虑的遗传特征是由两个基因 A 和 a 控制,那么就有三种基因型,表 1-2 给出父母基因型的所有可能组合使其后代形成每种基因对的概率。

表 1-1 投入产出分析表(单位:元)

		消耗部门			外界需求	总产出
		煤矿	电厂	铁路		
生产部门	煤矿	0	36506	15582	50000	102088
	电厂	25522	2808	2833	25000	56163
	铁路	25522	2808	0	0	28330
新创造的价值		51044	14041	9915		
总产出		102088	56163	28330		

表 1-2 基因型遗传概率

概率		父母					
		AA-AA	AA-Aa	AA-aa	Aa-Aa	Aa-aa	aa-aa
后代	AA	1	1/2	0	1/4	0	0
	Aa	0	1/2	1	1/2	1/2	0
	aa	0	0	0	1/4	1/2	1

设金鱼某种遗传病染色体的正常基因为 A,不正常基因为 a,那么 AA、Aa、aa 分别表示正常金鱼、隐性患者、显性患者。设初始分布为 90%正常金鱼,10%的隐性患者,无显性患者。考虑下列两种配种方案对后代该遗传病基因型分布的影响。

方案一:同类基因结合,均可繁殖;

方案二:显性患者不允许繁殖,隐性患者必须与正常金鱼结合繁殖。

解:设初始分布为 $x_1^{(1)}$, $x_2^{(1)}$, $x_3^{(1)}$,第 n 代分布为 $x_1^{(n)}$, $x_2^{(n)}$, $x_3^{(n)}$, $n=1, 2, \cdots$, 令

$$A=\begin{bmatrix}1 & 1/4 & 0\\ 0 & 1/2 & 0\\ 0 & 1/4 & 1\end{bmatrix},\ B=\begin{bmatrix}1 & 1/2 & 0\\ 0 & 1/2 & 0\\ 0 & 0 & 0\end{bmatrix},\ \boldsymbol{X}^{(n)}=\begin{bmatrix}x_1^{(n)}\\ x_2^{(n)}\\ x_3^{(n)}\end{bmatrix}$$

那么

$$\boldsymbol{X}^{(n)}=A^{n-1}\boldsymbol{X}^{(1)},\ \boldsymbol{X}^{(n)}=B^{n-1}\boldsymbol{X}^{(1)}$$

分别是两种情况下第 n 代的基因型分布。

```
》clear; A=[1  1/4  0; 0  1/2  0; 0  1/4  1];
》x=[0.9  0.1  0]';
》for i=2:20, x=A*x; end; x20=x
x20=
    0.9500
    0.0000
    0.0500
```

可见,按方案一,很多代以后,将出现 5%的稳定显性患者。若执行方案二

```
》clear; B=[1  1/2  0; 0  1/2  0; 0  0  0];
》x=[0.9; 0.1; 0];
```

```
》 for i = 2:20, x = B* x; end; x20 = x
x20=
    1.0000
    0.0000
         0
```

可知按方案二，很多代以后，不但不会出现显性患者，更令人鼓舞的是，连隐性患者也趋于消失。这个例子体现了杂交的优势。

现在我们用特征值和特征向量理论作进一步分析。

```
》clear; A = [1  1/4  0; 0  1/2  0; 0  1/4  1]; [P, T] = eig(A)
P=
    1.0000         0      -0.4082
         0         0       0.8165
         0    1.0000      -0.4082
T=
    1.0000         0            0
         0    1.0000            0
         0         0       0.5000
```

求得 3 个特征值 1，1，0.5，对应特征向量(1，0，0)′，(0，0，1)′，(−0.4082，0.8165，−0.4082)′。由于三个特征向量线性无关，从而 A 可相似对角化，即 $P^{-1}AP = T$。那么

$$A^n = (PTP^{-1})^n = PT^nP^{-1}$$

$$\lim_{n\to\infty}A^n = P(\lim_{n\to\infty}T^n)P^{-1} = P\begin{pmatrix}1 & & \\ & 1 & \\ & & 0\end{pmatrix}P^{-1}$$

```
》P* diag([1, 1, 0])* inv(P)
ans=
    1.0000    0.5000            0
         0         0            0
         0    0.5000       1.0000
```

对于任意初始分布 $x = (a, b, c)'$，$\lim\limits_{n\to\infty}x = (a+0.5b, 0, 0.5b+c)'$。这就从理论上证明了方案一最终导致隐性病患者消失。对于方案二，读者可自己作出类似的分析。

§1.5 实验习题

1 （方程组的解）解下列线性方程组，并总结线性方程组解的各种情况

(1) $\begin{pmatrix}4 & 1 & -1\\ 3 & 2 & -6\\ 1 & -5 & 3\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix} = \begin{pmatrix}9\\ -2\\ 1\end{pmatrix}$

(2) $\begin{pmatrix}4 & -3 & 3\\3 & 2 & -6\\1 & -5 & 3\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}=\begin{pmatrix}-1\\-2\\1\end{pmatrix}$

(3) $\begin{pmatrix}4 & 1\\3 & 2\\1 & -5\end{pmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix}=\begin{pmatrix}1\\1\\1\end{pmatrix}$

(4) $\begin{pmatrix}2 & 1 & -1 & 1\\1 & 2 & 1 & -1\\1 & 1 & 2 & 1\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix}=\begin{pmatrix}1\\2\\3\end{pmatrix}$，求通解。

2 （人口流动趋势）对城乡人口流动作年度调查，发现有一个稳定的朝向城镇流动的趋势，每年农村居民的5%移居城镇而城镇居民的1%迁出，现在总人口的20%位于城镇。假如城乡总人口保持不变，并且人口流动的这种趋势继续下去，那么一年以后住在城镇人口所占比例是多少？两年以后呢？十年以后呢？最终呢？

3 （经济预测）在某经济年度内，各经济部门的投入产出表如表1-3所示（单位：亿元）。假设 t 经济年度工业、农业及第三产业的最后需求均为17亿元，预测 t 经济年度工业、农业及第三产业的产出（提示：对于一个特定的经济系统而言，直接消耗矩阵和Leontief矩阵可视作不变）。

表1-3 投入产出表（单位：亿元）

投入部门	工业	农业	第三产业	最后需求	总产值
工业	6	2	1	16	25
农业	2.25	1	0.2	1.55	5
第三产业	3	0.2	1.8	15	20

4 （电路网）图1-1是连接三个电压已知终端的电路网，求 a、b、c 点的电压。

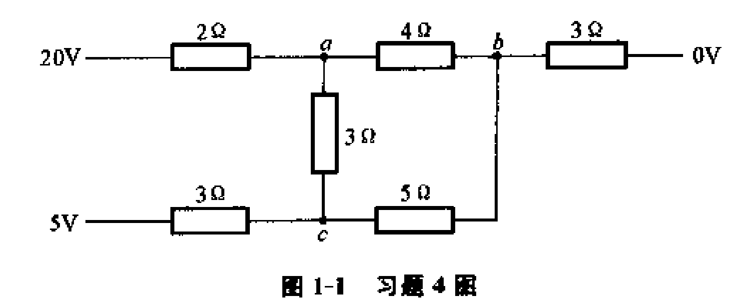


图1-1 习题4图

5 （二次型标准化）用正交变换化下列二次型为标准形

$$f(x_1, x_2, x_3)=x_1^2-4x_1x_2+4x_1x_3-2x_2^2+8x_2x_3-2x_3^2$$

6 （Hamilton-Carley定理）就矩阵 $A=\begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{pmatrix}$ 验证下列性质：

(i) 设 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 为 n 阶方阵 A 的特征值，则 $\sum_{i=1}^{n}\lambda_i = \sum_{i=1}^{n}a_{ii}$（$A$ 的迹），$\prod_{i=1}^{n}\lambda_i = (-1)^n \cdot |A|$；

(ii) 设 $f(x)$ 为 A 的特征多项式，则 $f(A) = 0$。

§1.6 补充知识:Gauss 消去法和病态方程病

1. 向量和矩阵的范数

我们知道，实数是有大小序关系的，复数本身没有大小序，但我们可用复数的模来定义它的大小，那么向量和矩阵怎样比较大小呢？这个问题无法从分量上回答。比如图 1-2，向量 $\boldsymbol{a} = (2, 2)$ 和 $\boldsymbol{b} = (3, 1)$，从第一分量比较，$\boldsymbol{a}$ 比 $\boldsymbol{b}$ 小，但从第二分量比较，$\boldsymbol{a}$ 比 $\boldsymbol{b}$ 大。

图 1-2 向量的范数

几何上，我们可通过 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的长度来比较。

$$\|\boldsymbol{a}\| = \sqrt{2^2+2^2} = \sqrt{8}$$

$$\|\boldsymbol{b}\| = \sqrt{3^2+1^2} = \sqrt{10}$$

从这一意义上，我们应该认为 $\boldsymbol{b}$ 较大。向量的大小一般用范数衡量。设向量 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$，常用的向量范数有

$$2\text{ 范数(即长度)}: \|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n}x_i^2}$$

$$1\text{ 范数}: \quad \|\boldsymbol{x}\|_1 = \sum_{i=1}^{n}|x_i|$$

$$\infty\text{范数}: \quad \|\boldsymbol{x}\|_\infty = \max_{1\leqslant i\leqslant n}|x_i|$$

虽然这些范数不尽相同，它们都是对分量大小的某种综合，不同范数间具有等价性。

范数的概念可推广到矩阵。

设 $A = (a_{ij})_{m\times n}$

2 范数 $\|A\|_2 = \sqrt{\max\limits_{1\leqslant k\leqslant n}(\lambda_k(A'A))}$，其中，$\lambda_k(A'A)$ 为 $A'A$ 的特征值

$$1\text{ 范数 } \|A\|_1 = \max_{1\leqslant j\leqslant n}\sum_{i=1}^{n}|a_{ij}|$$

$$\infty\text{范数 } \|A\|_\infty = \max_{1\leqslant i\leqslant m}\sum_{j=1}^{n}|a_{ij}|$$

使用 MATLAB 可求向量和矩阵各种范数。例如

```
》clear; A=[1 1/4 0; 0 1/2 0; 0 1/4 1]; x=[0.9, 0.1, 0];
》[norm(x), norm(x, 1), norm(x, inf)]      %分别为向量 x 的 2、1、∞范数
ans=
    0.9055      1.0000      0.9000
```

```
» [norm(A), norm(A, 1), norm(A, inf)]     % 分别为矩阵 A 的 2、1、∞范数
ans=
    1.0767    1.0000    1.2500
```

2. Gauss 消去法和选主元技术

线性方程组求解计算主要归结为 A 为可逆方阵时唯一解的求法。线性方程组数值解法主要分为两类:直接法和迭代法。直接法是基于矩阵分解的方法,包括 Gauss 消去法、LU 分解法、追赶法、平方根法(Cholesky 分解法)等。直接法求解精度高,是常用的方法。迭代法包括 Jacobi 迭代法、Gauss-Seidel 迭代法和超松弛(SOR)法等,但迭代法不一定收敛,主要用于一些特殊的大型稀疏型方程组。Gauss 消去法是先将方程化为三角形方程组,再回代求解,以三阶为例,

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 & \text{(a)} \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 & \text{(b)} \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 & \text{(c)} \end{cases}$$

(1) 消元过程　若 $a_{11} \neq 0$,则(b)-(a)$\times a_{21}/a_{11}$,(c)-(a)$\times a_{31}/a_{11}$,消去(b)、(c)中的 x_1 得

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 & \text{(a)} \\ \qquad a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 = b_2^{(1)} & \text{(d)} \\ \qquad a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 = b_3^{(1)} & \text{(e)} \end{cases}$$

若 $a_{22}^{(1)} \neq 0$,则(e)-(d)$\times a_{32}^{(1)}/a_{22}^{(1)}$,消去(c)式中的 x_2 得

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 & \text{(a)} \\ \qquad a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 = b_2^{(1)} & \text{(d)} \\ \qquad\qquad a_{33}^{(2)}x_3 = b_3^{(2)} & \text{(f)} \end{cases}$$

(2) 回代过程　若 $a_{33}^{(2)} \neq 0$,则由(f)式得 $x_3 = b_3^{(2)}/a_{33}^{(2)}$;将 x_3 代入(d)式得 $x_2 = (b_2^{(1)} - a_{23}^{(1)}x_3)/a_{22}^{(1)}$;将 x_2 和 x_3 代入(a)式得 $x_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11}$。

下列 M 函数 gauss.m 为简单 Gauss 消去法,并显示了消元过程。使用格式为

x = gauss(a, b)

其中 a 为系数方阵,b 为右端向量(或矩阵)。

%M 函数 gauss.m

```
function x = gauss(a, b)
[n, m] = size(b);
if size(a) = = [n, n]
    for i = 1:(n-1)
        if a(i, i) = = 0, error('Divide by zero'); end
b((i+1):n, :) = b((i+1):n, :) - a((i+1):n, i)*b(i, :)/a(i, i);
    a((i+1):n, i:n) = a((i+1):n, i:n) - ...
        a((i+1):n, i)*a(i, i:n)/a(i, i); [a, b]
```

```
        end
        if a(n, n) == 0, error('Divide by zero'); end
        x(n, :) = b(n, :)/a(n, n);
        for i = n-1:-1:1
            x(i, :) = (b(i, :) - a(i, (i+1):n)*x((i+1):n, :))/a(i, i);
        end
    else, error('Dimension of matrix must agree'), end
```

上面的简单 Gauss 消去法中，一旦消元过程中出现对角线上 $a_{kk}^{(k-1)}$（称为主元素）等于零，则计算无法进行，即使主元素不为零但其绝对值很小时，由于要用它作除数，容易造成其他元素数量级的巨大增长和舍入误差的扩散，从而使计算结果失真（见习题 8）。选列主元技术就是按列用绝对值最大的元素作为除数，以减少舍入误差的影响，具体地说就是在第 k 次消元前从 $a_{k,k}^{(k-1)}$，$a_{k+1,k}^{(k-1)}$，$\cdots$，$a_{n,k}^{(k-1)}$ 中选取绝对值最大的元素作为主元素，记为 $a_{pk}^{(k-1)}$。若 $p>k$，则把第 k 个方程与第 p 个方程互换，再按 Gauss 消去法进行消元。这样的方法称为**列选主元 Gauss 消去法**。MATLAB 中矩阵除法“A\b”对于 A 为普通方阵时，就采用这一方法计算。

3. 病态问题

考虑方程组

$$\begin{cases} 12x+35y=59 & \text{(A)} \\ 12x+35.000001y=59.000001 & \text{(B)} \end{cases} \tag{1.14}$$

解得 $x=2$，$y=1$。现在我们将 59.000001 改为 59（用以模拟数据误差、舍入误差等），方程组变为

$$\begin{cases} 12x+35y=59 & \text{(A)} \\ 12x+35.000001y=59 & \text{(C)} \end{cases} \tag{1.15}$$

解得 $x=59/12$，$y=0$（见图 1-3）。

图 1-3 病态方程组

我们看到右端不起眼的小扰动引起了解的大差异，注意这种差异并不是由算法引起（事实上两组解都是相应方程准确解），而与方程组性态有关（通常情况下数据上的小扰动不会引起这么大误差）。这样的方程组称为**病态问题**或**坏条件问题**，病态问题在线性代数的其他计算中（如求逆、特征值）也会出现。对于病态问题，特别要注意计算精度和数据精度。

方程组是病态程度的指标称为**条件数**，方阵的条件数定义为

$$\text{cond}(A)=\|A\|\,\|A^{-1}\|$$

其中 $\|\cdot\|$ 为矩阵的范数，最常用的是 2 范数（即 A^TA 的特征值最大值开方，也称为 A 的最大奇异值）。条件数越大，说明病态越严重。例如，对上述方程用 MATLAB 命令分析如下

```
» A = [12 35; 12 35.000001]; cond(A)
ans=
  2.2817e+008
```

可见 A 是病态矩阵。

4. 补充习题

7 对于 $k=10, 15, 16$，分别用不选主元的 Gauss 消去法和选主元的 Gauss 消去法(即 MATLAB 除法)解下列线性方程组，并用笔算分析产生误差的原因。

$$\begin{cases} 10^{-k}x_1 + x_2 = 1 \\ x_1 + x_2 = 2 \end{cases}$$

8 Hilbert 矩阵是著名的病态矩阵，n 阶 Hilbert 矩阵定义为 $A=(a_{ij})$，其中 $a_{ij}=1/(i+j-1)$，可以用 MATLAB 函数 hilb(n)产生。设 A 为一12 阶 Hilbert 矩阵，计算 cond(A)、A^{-1}、$A^{-1}A-E$ 及 $|A||A^{-1}|-1$，并分析结果。再用 Symbolic 命令验算。

实验二　购房贷款的利率
非线性方程和迭代

本实验中我们学习 MATLAB 有关非线性方程(组)求解的命令,加深对非线性函数和迭代等问题的理解,了解它们的一些应用。补充知识介绍了由非线性迭代产生的混沌现象。

§2.1　引例:购房贷款的利率

住房是居民消费的一个重要部分。对于大多数工薪阶层来说,用一次性付款买到称心如意的房子几乎是天方夜谭,所以大部分人选择银行按揭贷款,然后在若干年内分期还款。如果你借了 10 万,可别指望只还 10 万,因为你向银行借了钱,必然要按一定的贷款利率付给银行利息。房产广告五花八门,就是没有人告诉你贷款利率。你能计算出来吗?

这里是《新民晚报》2000 年 3 月 30 日第七版上的一则房产广告。不难算出,你向银行总共借了 25.2 万,30 年内共要还 51.696 万,约为当初借款的两倍。这个案例中贷款年利率是多少呢?

建筑面积	总　价	30%首付	70%按揭	月还款
85.98m²	36 万	10.8 万	30 年	1436 元

有人可能会这样计算

$$年利率 = (51.696 - 25.2)/30/25.2 = 3.5\%$$

但这个结果是错误的,因为你并不是等到 30 年后一次性还款。

设 x_k 为第 k 个月的欠款数,a 为月还款数,r 为月利率。我们得到下列迭代关系式

$$x_{k+1} = (1+r)x_k - a \tag{2.1}$$

那么

$$\begin{aligned} x_k &= (1+r)x_{k-1} - a = (1+r)^2 x_{k-2} - (1+r)a - a = \cdots \\ &= (1+r)^k x_0 - a[1 + (1+r) + \cdots + (1+r)^{k-1}] \\ &= (1+r)^k x_0 - a[(1+r)^k - 1]/r \end{aligned}$$

根据 $a = 0.1436$, $x_0 = 25.2$, $x_{360} = 0$ 得到

$$25.2(1+r)^{360} - 0.1436[(1+r)^{360} - 1]/r = 0 \tag{2.2}$$

这是一个关于月利率 r 的高次代数方程。从中解得 r,年利率 $R = 12r$。

§2.2 数学理论复习：非数性方程(组)

在科学研究和工程设计中常常遇到求解一非线性方程的问题，若方程是未知量 x 的多项式，称为高次**代数方程**；若方程包含 x 的超越函数，称为**超越方程**。一元非线性方程的一般形式为

$$f(x) = 0 \tag{2.3}$$

若对于数 α 有 $f(\alpha) = 0$，则称 α 为方程(2.3)的解或根，也称为函数 $f(x)$的零点，方程的根可能是实数也可能是复数，相应地称为**实根**和**复根**。如果对于数 α 有 $f(\alpha) = 0$，$f'(\alpha) \neq 0$ 则 α 称为**单根**，如果有 $k > 1$，$f(\alpha) = f'(\alpha) = \cdots = f^{(k-1)}(\alpha) = 0$ 但 $f^{(k)}(\alpha) \neq 0$，称为 **k 重根**，对于高次代数方程，其根的个数与其次数相同，如 4 次方程在复数范围内必有 4 个根(包括重数)，至于超越方程，其解可能是一个或几个甚至无穷多个，也可能无解。

常见的求解问题有如下两种要求，一种是要求定出在给定范围内的某个解，而解的粗略位置事先从问题的物理背景或应用其他方法得知，另一种是定出方程的全部解，或者给定区域内的所有解，而解的个数未知。除少数特殊的方程可以利用公式直接定出它的零点(如 4 次以下代数方程)，一般都没有解析求解方法，只能靠数值方法求得近似解。常见的数值方法有二分法、迭代法等。

n 元非线性方程组的一般形式为

$$f_i(x_1, x_2, \cdots, x_n) = 0, \ i = 1, \cdots, m \tag{2.4}$$

非线性方程组的解极少能用解析法求得。数值方法主要是综合运用线性方程组和非线性方程求解方法，常用方法是 Newton 法、拟 Newton 法、最优化方法等。

§2.3 数值解法：图解法和论代法

1. 图解法

适用于求一元或二元方程(组)低精度解或找迭代初值。

例 1 解方程

$$\sin(x) = 0.1x \tag{2.5}$$

解：由于 $-1 \leqslant \sin x \leqslant 1$，所以 $|x| \leqslant 10$，作出 $\sin(x) - 0.1x$ 在$[-10, 10]$范围内的图(图 2-1)，可看出根的大致位置。

```
» close; fplot('sin(x) - 0.1 * x', [-10, 10]);
» grid; zoom
```

可知±8.5，±7，±3，0 附近各有一个解。其中命令 zoom 使得图象可用鼠标点击放大，以提高局部观察精度。

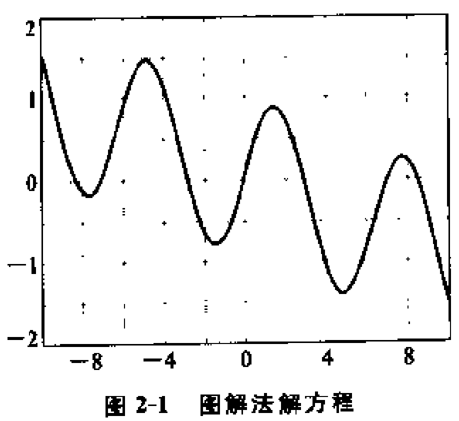


图 2-1 图解法解方程

例 2 用图解法解方程组

$$
\begin{cases} 4x_1 - x_2 + \dfrac{1}{10}e^{x_1} = 1 \\ -x_1 + 4x_2 + \dfrac{1}{8}x_1^2 = 0 \end{cases} \tag{2.6}
$$

解：本方程组可先化为一元方程求解，也可直接通过作二元函数图求解。下面使用后一方法。

```
》clear, close;
》x1a = -1:0.01:1; x2a = -1:0.01:1; [x1, x2] = meshgrid(x1a, x2a);
》f = 4*x1 - x2 + exp(x1)/10 - 1; g = -x1 + 4*x2 + x1.^2/8;
》contour(x1, x2, f, [0, 0]); % 曲面与平面 x2 = 0 的交线
》hold on; contour(x1, x2, g, [0 0]); hold off; zoom
```

可见在(0.25, 0.05)附近有一个解(见图 2-2)。

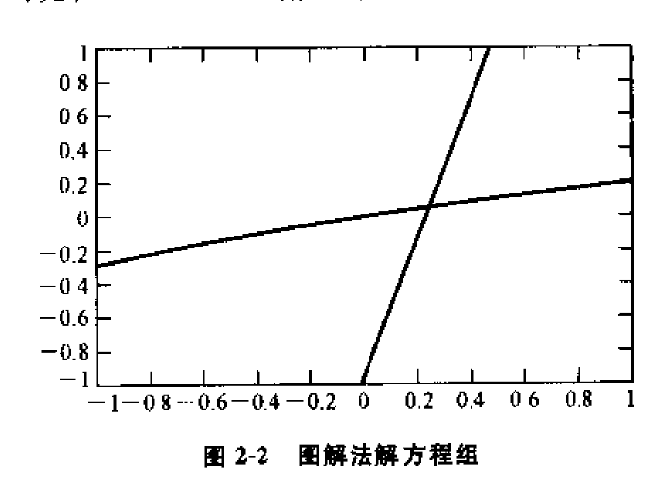


图 2-2　图解法解方程组

2. 迭代法

迭代法是从预知的解的初始近似值 x_0(简称初值)开始，利用某种迭代格式

$$x_{k+1} = g(x_k) \tag{2.7}$$

求得一近似值序列 x_1, x_2, …, x_k, x_{k+1}, …逐步逼近于所求的解 α(称为不动点)。这一方法是否成功取决于 3 个因素，首先 $x = g(x)$ 应与 $f(x) = 0$ 同解。其次初值 x_0 的选取是否合适，一般要与真解靠近。最后也是最关键的是迭代序列是否收敛。为了保证收敛性，在真解附近应有

$$|g'(x)| < 1$$

否则迭代序列可能产生复杂的性态(见补充知识)。

最常用的迭代法是 Newton 迭代法，其迭代格式为

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{2.8}$$

从几何上说 x_{k+1} 为用 $f(x)$在 x_k 处切线代替 $f(x)$求得的解，所以也称为**切线法**，当初值 x_0 与真解 α 足够靠近，Newton 迭代法收敛。对于单根，Newton 法收敛速度很快；对于重根，收敛较慢。

例 3　求下列方程的正根(要求精度 $\varepsilon = 10^{-6}$)

$$x^2 - 3x + e^x = 2 \tag{2.9}$$

解：令 $f(x) = x^2 - 3x + e^x - 2$，当 $x > 2$，$f(x) > 0$，$f'(x) > 0$ 即 $f(x)$单调上升，所以根在[0, 2]。我们先用图解法找初值，

```
》 fplot('x^2 - 3*x + exp(x) - 2', [0, 2]); grid on;
```

可见上述方程有唯一一个正根在 1 附近。取 $x_0 = 1$，迭代格式

$$x_{k+1} = x_k - \frac{x_k^2 - 3x_k + e^{x_k} - 2}{2x_k - 3 + e^{x_k}} \tag{2.10}$$

写出下列 M 脚本

```
%M 文件 eg2_3.m
clear; e = 1e - 6; format long;
x1 = 1
x0 = x1 + 2*e;      % 使 while 成立
while(abs(x0 - x1)>e)
    x0 = x1; x1 = x0 - (x0^2 - 3*x0 + exp(x0) - 2)/(2*x0 - 3 + exp(x0))
end; format
```

运行求得 $x = 1.44623868596643$。

§2.4 解方程和方程组的 MATLAB 命令

roots	求多项式的根;	fsolve	方程(组)数值求解;
fzero	求一元函数实根;	solve	符号方程(组)求解。

1. 多项式的根

roots(p) 可求得多项式 p 的所有复根。

MATLAB 中一个多项式用系数降幂排列向量来表示。例如,要求多项式 $x^3 + 2x^2 - 5$ 的根,用 MATLAB 命令

```
» roots([1 2 0 -5])
```

2. 一元函数零点

fzero 使用二分法和插值法求一元函数的零点,要求函数在所求零点附近变号。

fzero(F, X, tol),返回函数 F 的一个零点,其中 F 为字符串表示的函数或 M 函数名。X 为标量时,作为迭代初值;X 为向量[a, b]时,返回函数 F 在[a, b]中的一个零点,这时要求 F 在 a、b 两点异号。tol 为精度(缺省值 1e—4)。

为解例 1,

```
»fzero('sin(x) - 0.1*x', 6)
  ans=
      7.0682
»fzero('sin(x) - 0.1*x', [2, 6])
  ans=
      2.8523
```

注:fzero 只能求零点附近变号的根,读者试用 fzero 求解 $(x-1)^2 = 0$,看看发生了什么?

3. 非线性方程组求解

fsolve 是 MATLAB 优化工具箱(Optimzation Toolbox)里的函数,可求非线性方程或

多元非线性方程组的实根。用法与 fzero 类似。为了求解例 2,先写一个 M 函数 eg2_2fun.m

```
            %M 函数 eg2_2fun.m
        function y = fun(x)
        y(1) = 4*x(1) - x(2) + exp(x(1))/10 - 1;
        y(2) = - x(1) + 4*x(2) + x(1)^2/8;
```

然后用

```
»[x, y, f] = fsolve('eg2_2fun', [0, 0])      %x 返回解向量,y 返回误差向量,f > 0 则算法收敛
  x=
    0.2326      0.0565
```

或者直接用

```
»[x, y, f] = fsolve('[4*x(1) - x(2) + exp(x(1))/10 - 1, - x(1) + 4*x(2) + x(1).^2/8]', [0, 0])
```

求解。

注：fsolve 采用最小二乘优化法，稳定性比 fzero 好，但 fsolve 可能陷入局部极小。试用 fsolve 解 $x^2+x+1=0$，看看发生了什么？告诉你，不要完全相信计算机！对于 fzero 和 fsolve 结果最好通过直接计算函数值验证一下。

4. 解析求解 solve

solve 是符号数学工具箱中一个功能强大的解方程(组)命令。可求各种类型方程(组)的解析解。当找不到解析解时，solve 会自动寻求一个近似解，且精度很高。例如用

```
»solve('a*x^2 + b*x + c', 'x')
```

可得二次多项式求根公式；用

```
»solve('x^2 - 3*x + exp(x) - 2', 'x')
```

可求解例 3；用

```
»[x, y] = solve('4*x - y + exp(x)/10 = 1', '- x + 4*y + x^2/8 = 0', 'x, y')
```

可求解例 2。注意所得解与 fsolve 的结果不同。

注：虽然 solve 可用于求数值解，但速度很慢，且有很大局限性，我们不提倡使用。

§2.5 实验例题

例 4 （贷款年利率）考虑方程(2.2)。常识上，r 应比当时活期存款月利率略高一些。我们用活期存款月利率 0.0198/12 作为迭代初值，用 fzero 求解

```
»r = fzero('25.2*(1 + x)^360 - ((1 + x)^360 - 1)/x*0.1436', 0.0198/12);
»R = 12*r
R=
  0.0553
```

得到年利率 5.53%。事实上，本案是公积金和商业性组合贷款的平均利率。

注：在使用 fzero, fsolve 等命令时，若把函数直接写在命令中，自变量必须用 x。若用 M

函数表示，就灵活得多。许多 MATLAB 命令都有类似的问题。

例 5 （拴牛鼻的绳子）农夫老李有一个半径 10m 的圆形牛栏，里面长满了草，老李要将家里一头牛拴在一根栏桩上，但只让牛吃到一半草，他想让上大学的儿子告诉他，拴牛鼻的绳子应为多长？

解：这个问题如图 2-3 所示。设 A 为栏桩，绳 AB 长 R，半径 $OA = OB = r$，$\theta =$ 角 OAB，那么 $R = 2r\cos\theta$。扇形 BAC 面积

$$S_1 = 2\pi R^2\theta/(2\pi) = \theta R^2$$

冠形 ADB 圆积

$$S_2 = (\pi - 2\theta)\pi r^2/(2\pi) - Rr(\sin\theta)/2$$
$$= (\pi/2 - \theta)r^2 - Rr(\sin\theta)/2$$

那么

$$\theta R^2 + (\pi - 2\theta)r^2 - Rr(\sin\theta) = \pi r^2/2$$

即

$$\sin(2\theta) - 2\theta\cos(2\theta) = \pi/2 \quad (2.11)$$

图 2-3 例 5 图

从方程(2.11)解出 θ，即可求得 R。先写 M 函数

```
%M 函数 eg2_5fun.m
function y = f(theta)
y = sin(2*theta) - 2*theta*cos(2*theta) - pi/2;
```

再用

```
》theta = fzero('eg2_5fun', pi/4); R = 20*cos(theta)
```

得 R = 11.5873。

§2.6 实验习题

1 （多项式方程的根）4 次以下的多项式方程有公式解，试用 solve 解下列方程

$$ax^2 + x + 1 = 0$$
$$ax^3 + x + 1 = 0$$
$$ax^4 + x + 1 = 0$$
$$ax^5 + x + 1 = 0$$

再对 $a = 1$ 时，用 roots 或 solve 求解。

2 （超越方程）想越方程的解有时是很复杂的，作出

$$f(x) = x\sin(1/x)$$

在[-0.1, 0.1]内的图，可见在 $x = 0$ 附近 $f(x) = 0$ 有无穷多个解，并设法求出它的解。

3 （月还款额）作为房产公司的代理人，你要迅速准确回答客户各方面的问题。现在有个客户看中了贵公司一套建筑面积为 120m²，单价 5200 元/m² 的房子。他计划首付 30%，其余 70%用 20 年按揭贷款（贷款年利率 5.58%）。请你提供下列信息：房屋总价格、首付款

额、月付还款额。

4 （线性迭代）迭代过程

$$x_{k+1} = g(x_k)$$

的收敛性主要条件是在根的附近满足 $|g'(x)| < 1$。从理论上证明线性迭代

$$x_{k+1} = ax_k + 1$$

只有两种极限形态:不动点或无穷大。分别就 $a = 0.9, -0.9, 1.1, -1.1$（取 $x_0 = 1$，迭代 20 步）用图形显示迭代过程的不同表现（提示:用 subplot 将 4 个子图放在一个图形窗口比较）。

5 （弦截法）Newton 迭代法是一种速度很快的迭代方法，但是它需要预先求得导函数。若用差商代替导数，可得下列弦截法

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k)$$

这一迭代法需要两个初值 x_0，x_1，编写一个通用的弦截法计算机程序并用以解例 3。

6 （椭圆的交点）两个椭圆可能具有 0～4 个交点，求下列两个椭圆的所有交点坐标

$$(x-2)^2 + (y-3+2x)^2 = 5$$
$$2(x-3)^2 + (y/3)^2 = 4$$

7 （化学反应平衡）一等克分子数一氧化碳（CO）和氧气（O_2）的混合物在 300K 和 5bar（1bar $= 10^5$Pa）压力下达到平衡，理论反应方程式为

$$CO + 0.5O_2 \rightarrow CO_2$$

实际反应方程式为

$$CO + O_2 \rightarrow xCO + 0.5(1+x)O_2 + (1-x)CO_2$$

剩余 CO 比值 x 满足化学平衡方程式

$$K_p = \frac{(1-x)\sqrt{10.52+x}}{x\sqrt{1+x}\sqrt{p}}, \quad 0 < x < 1$$

这里 $K_p = 3.06$，$p = 5$bar，求 x.

§2.7 补充知识：认识混沌

线性迭代要么收敛于它的不动点，要么趋于无穷大（见习题 4）；而不收敛的非线性迭代可能会趋于无穷大，也可能趋于一个周期解，但也有可能在一个有限区域内杂乱无章地游荡，这类由确定性运动导致的貌似随机的现象称为**混沌现象**。下面我们就 Logistic 迭代来研究这一现象。

1. 昆虫数量的 Logistic 模型

$$x_{k+1} = ax_k(1-x_k), \quad 0 \leqslant a \leqslant 4 \tag{2.12}$$

x_k 表示第 k 代昆虫数量（1 表示理想资型环境最大可能昆虫数量）。（2.12）式反映了下一代对上一代的既依赖又竞争的关系。当上一代很少，繁殖能力不够，从而后代很少；当上一代

很多，会吃掉很多食物，后代难以存活，从而后代也很少。a 为资源系数，$0 \leqslant a \leqslant 4$ 保证了 x_k 在区间(0, 1)上封闭。

2. 平衡与稳定

称 α 为映射 $g(x)$ 的**平衡解或不动点**，若 $g(\alpha) = \alpha$。对于(2.12)式，$g(x) = ax(1-x)$。解方程

$$x = ax(1-x)$$

得(2.12)式两个不动点 0 和 $1-1/a$。若初始值恰好为不动点，迭代(2.12)的值永不会改变。如果对于 x_0 不动点附近的初始值 x_0，(2.12)式收敛于此不动点，我们就称这一不动点是稳定的。

当 $0 \leqslant a < 1$，在[0, 1]内只有一个不动点 0，且由 $|g'(0)| = a < 1$，可知它是稳定的。说明资源匮乏时，昆虫趋于消亡。

当 $a > 1$，不动点 0 不再稳定，而由 $|g'(1-1/a)| = |2-a| < 1$ 可知 $1 < a < 3$ 时不动点 $1-a^{-1}$ 稳定，说明资源适当时，昆虫稳定于一定数量。

3. 周期解、分叉和混沌

称 α 为映射 $g(x)$ 的**周期 k 点**，若 $g^k(\alpha) = \alpha$，而对任意 $j < k$，$g^j(\alpha)$ 不等于 α(这里 g^j 表示 g 的 j 次复合)。并称 α, $g(\alpha)$, …, $g^{k-1}(\alpha)$为**周期 k 轨道**。

让我们来求(2.12)式的周期 2 轨道，

```
》solve('x-a*a*x*(1-x)*(1-a*x*(1-x))=0')
ans=
[                                      0]
[                               (-1+a)/a]
[(1/2*a+1/2+1/2*(a^2-2*a-3)^(1/2))/a]
[(1/2*a+1/2-1/2*(a^2-2*a-3)^(1/2))/a]
```

可见当 $a^2-2a-3>0$，即 $a>3$，出现两个周期 2 解，可以证明 $3<a<1+\sqrt{6}$ 时周期 2 轨道稳定(习题 8)。迭代开始发生所谓**倍周期分岔**，从周期 2，周期 4，…，周期 2^n，…直到 $a_\infty = 3.569945672\cdots$。说明 a 在$[3, a_\infty]$取值时，昆虫数量呈现规律性振荡。当 $a > a_\infty$，(2.12)式的迭代序列几乎杂乱无章，即所谓**混沌**。我们通过下列例子形象地显示上述现象。

例 6 (分叉图)对于 a 在[0, 4]的不同值，画出 Logistic 迭代的极限形态图。

下列 M 文件 eg2_6.m 对于每一个 a 值，随机产生一个初始值。文件显示了前 20 步迭代的变化。最后用第 180～200 步迭代值表示极限形态，见图 2-4。

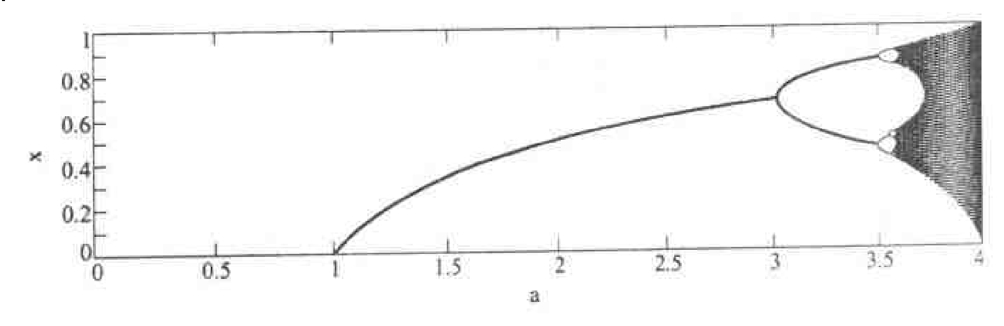


图 2-4 混沌分叉图

```
%M 脚本 eg2_6.m
clear; close; a = 0:0.01:4;
M = length(a); K = 200; x = zeros(K, M); x(1, :) = rand(1, M);
for m = 1:M, for k = 1:K - 1
    x(k + 1, m) = a(m)* x(k, m)* (1 - x(k, m));
end, end
for k = 1:20,
  plot(a, x(k, :), '.'); title(['k = ', int2str(k)]); pause(2);
end;
plot(a, x(180:K, :), '.'); xlabel('a'); ylabel('x'); hold off;
```

4. 混沌的特征

混沌是由确定性系统产生的貌似随机的现象。一般认为，混沌有下列几个主要特征

(i) **初值敏感性**：两个任意近的点出发的两条轨迹迟早会分得很开；

(ii) **遍历性**：任意点出发的轨迹总会进入[0, 1]内任意小的开区间。

例 7 （*初值敏感性*）下列 M 文件 eg2_7.m 验证了 Logistic 迭代序列的初值敏感性。对于靠得很近的两个初值（相差仅 1e−4），我们画出了两个序列 50 步内的误差图（图 2-5）。可见 10 步以后，差异增大，有时甚至接近 1。

```
%M 脚本 eg2_7.m
clear; close; a = 4; e = 1e - 4;
x = zeros(50, 2); x(1, :) = [0.4, 0.4 + e];
for i = 2:50
    x(i, :) = a* x(i - 1, :).* (1 - x(i - 1, :));
end
y = x(:, 1) - x(:, 2); plot(y)
```

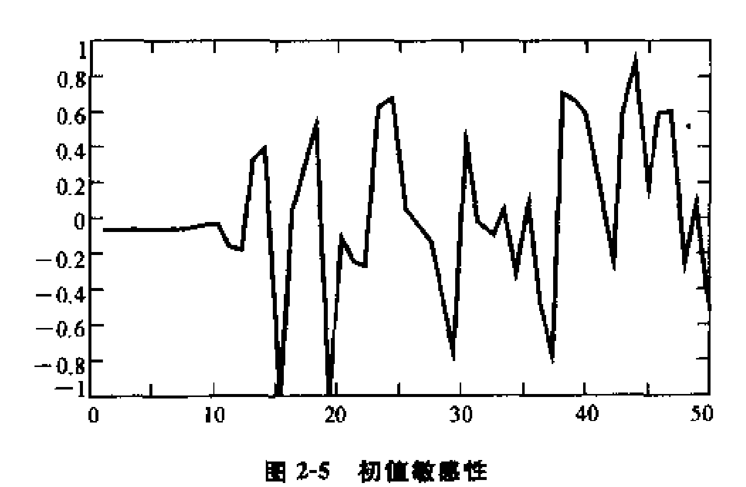


图 2-5 初值敏感性

例 8 （*蛛网图*）我们用蛛网图来显示混沌的遍历性。将(2.12)式改写为平面迭代

$$y_k = ax_k(1 - x_k),$$

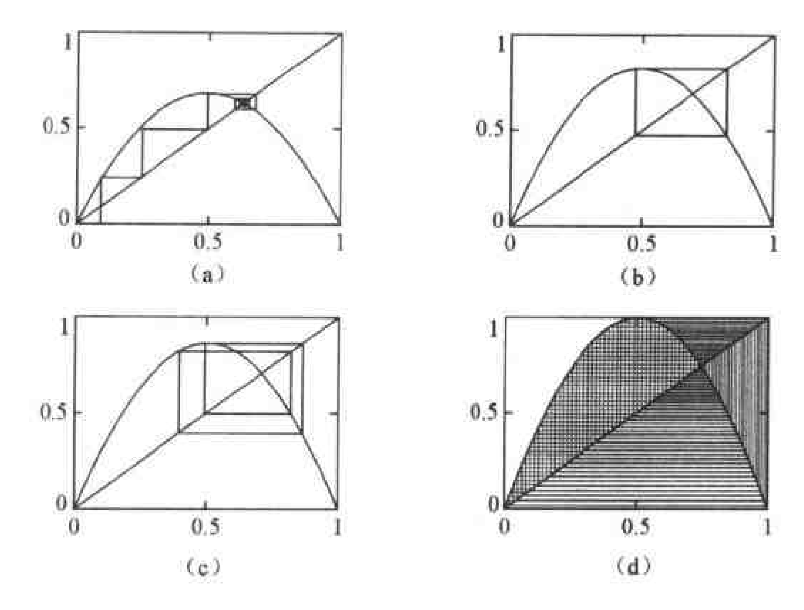
$$x_{k+1} = y_k$$

蛛网图正好显示迭代计算 x_0，y_0，x_1，y_1，…的一系列变化过程。下列 M 函数 eg2_8.m 是一个通用的 Logistic 蛛网图函数。作出系数为 a，初值为 x_0，从第 m 步到第 n 步的迭代过程。

```
%M 函数 eg2_8.m
function f = eg2_8(a, x0, m, n)
x = 0:0.01:1; y = a*x.*(1-x);
plot(x, x, 'r', x, y, 'r'); hold on;
clear x, y;
x(1) = x0;
y(1) = a*x(1)*(1-x(1));
x(2) = y(1);
if m<2, plot([x(1), x(1), x(2)], [0, y(1), y(1)]); end
for i = 2:n
    y(i) = a*x(i)*(1-x(i));
    x(i+1) = y(i);
    if i>m, plot([x(i), x(i), x(i+1)], [y(i-1), y(i), y(i)]); end
end
hold off;
```

在命令窗口执行

```
》subplot(2, 2, 1); eg2_8(2.7, 0.1, 1, 100);    %收敛迭代
》subplot(2, 2, 2); eg2_8(3.4, 0.1, 50, 500);   %周期 2
》subplot(2, 2, 3); eg2_8(3.5, 0.1, 50, 500);   %周期 4
》subplot(2, 2, 4); eg2_8(4, 0.1, 50, 500);     %混沌
```



(a)　(b)　(c)　(d)

图 2-6　蛛网图

(a) 收敛；(b) 周期 2；(c) 周期 4；(d) 混沌

可见混沌迭代对于初值为 0.1，轨迹遍历了[0, 1]区间(图 2-6)。

5. 补充习题

8　证明，当且仅当 $3 < a < 1+\sqrt{6}$，Logistic 映射的周期 2 轨道是稳定的。

9　作出习题 4 的蛛网图。

10　用数值方法分析何时出现周期 4 轨道，又于何时失稳？然后再用理论分析求解。比较结果。

11　（Henon 吸引子）混沌和分形的著名例子，迭代模型为

$$\begin{cases} x_{k+1} = 1 + y_k - 1.4x_k^2 \\ y_{k+1} = 0.3x_k \end{cases}$$

取初值 $x_0 = 0$，$y_0 = 0$，进行 3000 次迭代，对于 $k > 1000$，在 (x_k, y_k) 处亮一点（注意不要连线）可得所谓 Henon 引力线图。

实验三　最佳订货量　极限、导数和极值

本实验中我们学习 MATLAB 有关极限、级数、导数和极值等命令，加深对微分学一些基本概念及理论的理解，讨论最佳定货量和银行复利等应用问题。补充知识介绍了数值计算法求解数学问题的局限性和最优化求解的黄金分割法。

§3.1　引例：最佳订货最问题

汽车工厂为了保证生产的正常运作，配件供应一定要有保障。这些配件并不是在市场上随时可以买到的，所以往往要预先从配件供应商那里订货。由于配件供应商并不是生产单一产品，为你的订货必须要在流水线上作出调整，所以每次订货需要收取一定量的生产准备费。配件供应商的生产能力很大，开工后很快可以生产许多配件，但是你的汽车工厂并不是立即需要这么多，往往要在仓库里储存一段时间，为此你要付出储存费。如果订货量很小，必然需要频繁定货，造成生产准备费的增加；反之，若订货量很大，定货周期必然延长，生产准备费下降，但这样会造成储存费的增加。如何确定合适的订货量?

实践中，这是一个相当复杂的问题，因为在多方面会受到市场波动的影响。我们先作一些必要的假设将问题简化。

(1) 汽车工厂对配件的日需求量是恒定的，每日为 r 件；

(2) 所订配件按时一次性交货，生产准备费每次 k_1 元；

(3) 储存费货当日实际储存量计算，储存费每日每件 k_2 元；

(4) 你的工厂不允许缺货。

设一次订货 x 件，由于工厂不允许缺货，商为了节省储存费，交货日期应定为恰好用完时，所以订货周期

$$T = x/r \tag{3.1}$$

由于日需求量是恒定的，可以计算出第 t 天的储存量为

$$q(t) = x - rt,\ 0 < t < T \tag{3.2}$$

由于第 t 天的储存费为 $k_2q(t)$，一个周期的总储存费

$$k_2\sum_{t=1}^{T}q(t) \approx k_2\int_0^T q(t)\,\mathrm{d}t \tag{3.3}$$

根据(3.1)式、(3.2)式和(3.3)式，得到一个周期总费用

$$C(x) = k_1 + k_2\frac{x^2}{2r}$$

优化目标是使单位产品费用

$$f(x) = \frac{C(x)}{x} = \frac{k_1}{x} + \frac{k_2 x}{2r}$$

达到最小。由 $f'(x) = 0$ 即

$$-\frac{k_1}{x^2} + \frac{k_2}{2r} = 0$$

可解得 $x = \sqrt{\frac{2k_1 r}{k_2}}$，这就是著名的**经济批最订货公式**。

§3.2 数学理论复习：数分学

1. 极限和连续

极限是高等数学最基本的概念，它带来了很多深刻的结果。

数列极限：如果对于 $\forall \varepsilon > 0$，存在正整数 N，使当 $n > N$ 时有

$$| x_n - a | < \varepsilon \tag{3.4}$$

则称 a 为数列 x_n 的极限，或称 x_n 收敛于 a。记为 $\lim\limits_{n\to\infty} x_n = a$ 或 $x_n \to a$。直观上表示：n 趋无穷大时，x_n 无限接近 a。

函数极限：如果当 $x \to x_0$ 时有 $f(x) \to A$，则称 A 为函数 $f(x)$ 当 $x \to x_0$ 时的极限。记为 $\lim\limits_{x\to x_0} f(x) = A$。若仅当 $x \to x_0$ 且 $x > x_0$（或 $x < x_0$）时有 $f(x) \to A$，则称 A 为 $f(x)$ 当 $x \to x_0$ 时的右极限（或左极限），记为 $f(x_0+0)$（或 $f(x_0-0)$）。当 $f(x_0+0) = f(x_0-0)$，$f(x)$（当 $x \to x_0$ 时）的极限存在且等于这个值。

连续：若 $f(x_0+0) = f(x_0)$（或 $f(x_0-0) = f(x_0)$），则 $f(x)$ 在 x_0 右连续（或左连续）。若 $f(x)$ 在 x_0 右连续且左连续，则称 $f(x)$ 在 x_0 **连续**。若 $f(x)$ 在区间 (a, b) 内每一点都连续，则称 $f(x)$ 在开区间 (a, b) 连续。进一步，若 $f(x)$ 还在 a 右连续而 b 左连续，则称 $f(x)$ 在闭区间 $[a, b]$ 连续。**连续函数在则区间上必然能达到最大值和最小值，且可取得最大值和最小值间的任意值。**

2. 微分与导数

设 x 与 y 是相关联的两个变量，用函数表示为 $y = f(x)$。对于 x 的一个小增量 $\Delta x = x - x_0$（称为**差分**），引起 y 的一个小增量 $\Delta y = f(x) - f(x_0)$，若

$$\Delta y = A\Delta x + o(\Delta x) \tag{3.5}$$

其中 A 是不依赖 Δx 的常数，而 $o(\Delta x)$ 是 Δx 的高阶无穷小量（即 $o(\Delta x)/\Delta x \to 0$），那么称 $f(x)$ 在 x_0 **可微**，并记为

$$\mathrm{d}y = A\mathrm{d}x \tag{3.6}$$

其中 $\mathrm{d}x$，$\mathrm{d}y$ 分别称为 x 和 y 的**微分**。

函数 $f(x)$ 在点 $x = x_0$ 的导数定义为

$$f'(x_0) = \lim_{h\to 0} \frac{f(x_0+h) - f(x_0)}{h} \tag{3.7}$$

它反映了在 x_0 点附近函数 $f(x)$ 的变化率。当 $f'(x_0) > 0$，函数在 x_0 点附近是上升的，反之 $f'(x_0) < 0$，函数在 x_0 点附近是下降的，而当 $f'(x_0) = 0$，往往（但不一定）标志函数在 x_0 点达到**局部极大或局部极小**。函数在 x_0 点达到**局部极大**（或局部极小）的充分条件是 $f'(x_0) = 0$ 且 $f''(x_0) < 0$（或 $f''(x_0) > 0$）。从几何意义上说，$f'(x_0)$ 是函数在点 x_0 切线的斜率。显然有 $f'(x) = \frac{\mathrm{d}f}{\mathrm{d}x}$，可见导数是微分的商，所以也称**微商**。

Taylor 公式是微分学非常重要的一个结论。当 $f(x)$ 在含有 x_0 某个开区间内具有直到 $n+1$ 阶的导数，那么当 $x \in (a, b)$ 有

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 +$$

$$\cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1} \tag{3.8}$$

其中 ξ 是 x_0 与 x 之间某个值。Taylor 公式表明一个可微性很好的函数可局部地用多项式函数近似地代替。

特别地，当 $n = 0$ 得**微分中值定理**

$$f(x) - f(x_0) = f'(\xi)(x - x_0) \tag{3.9}$$

它表明在 x 与 x_0 之间存在一点 ξ，使 $f'(\xi)$ 恰为 $f(x)$ 从 x_0 到 x 的平均变化率，但中值定理不能给出 ξ 的确切位置。当 x 离 x_0 不远，且 $f'(x)$ 在 x_0 附近连续，有

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \tag{3.10}$$

它表明任意光滑函数可局部线性化，常用于非线性函数的近似分析和计算。

3. 多元函数微分学

极限、**连续**、**微分**、**导数**等概念容易推广到多元函数。设二元函数 $f(x, y)$ 在点 (x_0, y_0) 附近有定义，当 (x, y) 以任何方式趋向于 (x_0, y_0) 时，$f(x, y)$ 趋向于一个确定的常数 A，则称 A 为 $f(x, y)$ 当 $x \to x_0$，$y \to y_0$ 时的二重极限，记为

$$\lim_{\substack{x \to x_0 \\ y \to y_0}} f(x, y) = A \tag{3.11}$$

若 $\lim_{\substack{x \to x_0 \\ y \to y_0}} f(x, y) = f(x_0, y_0)$，称 $f(x, y)$ 在 (x_0, y_0) 点**连续**。二元函数 $f(x, y)$ 在点 (x_0, y_0) 的关于变量 x 和 y 的**偏导数**分别定义为

$$\begin{aligned} f'_x(x_0, y_0) &= \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x, y_0) - f(x_0, y_0)}{\Delta x} \\ f'_y(x_0, y_0) &= \lim_{\Delta y \to 0} \frac{f(x_0, y_0 + \Delta y) - f(x_0, y_0)}{\Delta y} \end{aligned} \tag{3.12}$$

分别记为 $\left.\frac{\partial f}{\partial x}\right|_{(x_0, y_0)}$ 和 $\left.\frac{\partial f}{\partial y}\right|_{(x_0, y_0)}$。二元函数在 (x_0, y_0) 附近变化的性态主要由 $\left.\frac{\partial f}{\partial x}\right|_{(x_0, y_0)}$ 和 $\left.\frac{\partial f}{\partial y}\right|_{(x_0, y_0)}$ 共同决定，它们合称为**梯度**，记为 $\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$。$f(x, y)$ 在 (x_0, y_0) 取得局部

极大或极小的必要条件(但非充分条件)是

$$\nabla f \mid_{(x_0, y_0)} = 0 \tag{3.13}$$

充分条件是(3.13)式且下列 Hesse 矩阵负定(局部极大)或正定(局部极小)

$$\begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}$$

二元函数也有类似的 Taylor 公式。特别地有,在(x_0, y_0)附近

$$f(x, y) \approx f(x_0, y_0) + f'_x(x_0, y_0)(x - x_0) + f'_y(x_0, y_0)(y - y_0) \tag{3.14}$$

它将二元函数在(x_0, y_0)附近局部线性化。

§3.3 数 值 数 分

由导数的定义,若 $f(x)$在 $x = a$ 可导,设 $h > 0$ 且足够小,由(3.7)式的右极限得

$$f'(a) \approx \frac{f(a+h) - f(a)}{h} \tag{3.15}$$

由(3.7)式的左极限得

$$f'(a) \approx \frac{f(a) - f(a-h)}{h} \tag{3.16}$$

分别称为**向前差商**和**向后差商**。事实上,对于连续函数,两式正好求得右导数和左导数。两式平均得

$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h} \tag{3.17}$$

称为**中心差商**。中心差商精度较高。

高阶导数也可用差商法求得,例如 3 点二阶导数公式为

$$f''(a) \approx \frac{f(a+h) - 2f(a) + f(a-h)}{h^2} \tag{3.18}$$

§3.4 求值限、导数和值值的 MATLAB 命令

limit	符号命令求极限;	taylor	符号 Taylor 展开;
symsum	符号级数求和;	polyder	多项式求导;
diff	数值差分或符号求导;	fmin	一元函数极值;
gradient	数值梯度;	fmins	多元函数极值。

1. 数值求导

> dx = diff(x)　返回向量 x 的差分；
> Fx = gradient(F, x)　返回向量 F 表示的一元函数沿 x 方向的数值梯度（即导函数 $F'(x)$），其中，x 是与 F 同维数的向量；
> [Fx, Fy] = gradient(F, x, y)　返回矩阵 F 表示的二元函数的数值梯度（F'_x，F'_y），当 F 为 $m \times n$ 矩阵时，x、y 分别为 n 维和 m 维的向量。

用 MATLAB 数值求导最经济的方法是将其处理成差分的商。例如

```
»clear; x = [1  1.1  1.2  1.3]; y = x.^3;
»dy = diff(y)./diff(x)
dy=
  3.3100  3.9700  4.6900
```

求得 y′(1)，y′(1.1)和 y′(1.2)的近似值(向前差商)。若用梯度求解

```
»dy = gradient(y, x)
dy=
    3.3100  3.6400  4.3300  4.6900
```

求得 y′(1)，y′(1.1)，y′(1.2)和 y′(1.3)的近似值。准确解

```
»3* x.^2
ans=
  3.0000  3.6300  4.3200  5.0700
```

gradient 使用中心差商，从而误差较小。

2. 函数极值

> x = fmin('fun', a, b)　求一元函数 $y = f(x)$ 在$[a, b]$内的局部极小值点。这里 fun 可直接用字符串表示，也可以是 M 函数名。采用黄金分割法和抛物线插值法。
> x = fmins('fun', x0)　求多元函数 $y = f(x)$ 在 x_0 出发的局部极小值点，这里 x，x_0 均为向量。采用 Nelder-Meade 单纯形搜索法。

例 1　求二元函数 $f(x, y) = 5 - x^4 - y^4 + 4xy$ 在原点附近的极大值。

解：问题等价于求 $-f(x, y)$ 的极小值。

```
»fun = 'x(1)^4 + x(2)^4 - 4* x(1)* x(2) - 5';
»x = fmins(fun, [0, 0]), f = - eval(fun)    % eval(fun)执行字将串 fun 表达的运算
x=
  1.0000  1.0000              %极大值点
f=
  7.0000                      %极大值
```

注：在使用 fmins 等命令时，若把函数直接写在算式中，自变量必须用 x(1)，x(2)，…。

3. 解析运算

limit，symsum，diff，taylor 都是符号运算命令。

```
limit(s, x, a)  返回符号表达式 s 当 x → a 时的极限。
symsum(s, n, a, b)  返回符号表达式 s 表示的通项当自变量 n 由 a 到 b 的和。
diff(s, x, n)  返回符号表达式 s 对 x 的 n 阶导数。
taylor(s, n, a)  返回符号表达式 s 在 a 点 Taylor 展开到 n 次式。
```

例如，

```
》syms x; taylor(sin(x), 6, 0)
ans=
x-1/6*x^3+1/120*x^5
```

例 2 计算 $\lim\limits_{n\to\infty}\left(1+\frac{x}{n}\right)^n$，$\sum\limits_{n=1}^{\infty}(-1)^n\frac{x^n}{n}$，$\frac{\partial^2}{\partial x^2}[\sin(xy)]$

```
》syms n x y;
》limit((1+x/n)^n, n, inf)
ans=
exp(x)
》symsum((-1)^n*x^n/n, n, 1, inf)
ans=
-log(1+x)
》diff(sin(x*y), x, 2)
ans=
-sin(x*y)*y^2
```

§3.5 实验例题

例 3 (不同期限的存款利率)若银行一年定期年利率为 r，那么储户存 1 万元钱，一年到期后结算额为 $1+r$ 万元；若三月定期年利率也为 r，每三月储算一次，由于复利，储户存 1 万元钱一年后可得 $(1+r/4)^4$ 万元，显然

$$(1+r/4)^4 > 1+r$$

这是由于多次结算增加了复利。进一步，每月结算一次，总结算额为 $(1+r/12)^{12}$ 万元，每天结算一次，总结算额为 $(1+r/365)^{365}$ 万元。结算越频繁，获利越大。现在我们已进入了电子商务时代，允许储户随时存款或取款，如果一个储户连续不断存款取款，结算频率趋于无穷大，这量味银行要不断地向顾客付利息，称为连续复利。连续复利会造成总结算额无限增大吗？如果活期存款年利率为 1.98%，那么一年、三年、十年定期存款的年利率应定为多少才是等价的？

解：一般地，设若结算频率为 n，年利率为 x，第 k 次结算额为 a_k，那么得到下列差分方程

$$a_k = (1+x/n)a_{k-1},\ a_0 = 1$$

则一年总结算额为 $a_n = (1+x/n)^n$，由于

$$\lim_{n\to\infty}(1+x/n)^n = e^x$$

可见总结算额有一个上限 e^x。在这个例子中，尽管 $n\to\infty$ 极限过程仅是理论上的，但其结果比一个具体的 n 更重要。它表明 n 足够大时，结果将稳定于这个值。事实上，$n=365$ 时，总结算额已与 e^x 相当接近。

我们把活期存款利率作为连续复利率 $r_0 = 1.98\%$。设一年定期利率为 r，那么应有

$$1+r = e^{r_0}$$

从而 $r = e^{r_0}-1 = 2\%$。同理三年定期利率 $r = (e^{3r_0}-1)/3 = 2.04\%$，十年定期利率 $r = (e^{10r_0}-1)/10 = 2.19\%$。银行实际定期年利率要更高，以鼓励长期存款。

例 4 （导函数、单调性和极值点）导函数的值能反映函数的变化。当 $f'(x_0)>0$，函数在 x_0 点附近是上升的，反之 $f'(x_0)<0$，函数在 x_0 点附近是下降的，而当 $f'(x_0)=0$，往往（但不一定）标志函数在 x_0 点达到局部极大或局部极小。下面我们通过图象来认识。

考虑函数 $f(x) = x^2\cos(x^2+3x-4)$ 在[−2, 2]内的图象，

```
» fun='x*x*cos(x*x+3*x-4)'; fplot(fun, [-2, 2]); grid;
```

可以明显看到 $f(x)$ 在 0.5 和 1.5 附近各有一个局部极小值点，在 1 附近有一个局部极大值点，在[-0.5, 0.5]有一个不明朗地带。

现在我们来看导函数。

```
» dfun=diff(fun)
ans=
2*x*cos(x^2+3*x-4)-x^2*sin(x^2+3*x-4)*(2*x+3)
```

让我们作出导函数图象，并与 $f(x)$ 图象叠在一起（图 3-1）。

```
»dfun=char(dfun);
»hold on; fplot(dfun, [-2, 2], 'r'); hold off;
```

通过图 3-1，可以直观看到 $f'(x)$ 与 $f(x)$ 的关系。进一步我们考虑区间[−0.5, 0.5]。

```
»fplot(dfun, [-0.5, 0.5], 'r'); grid;
```

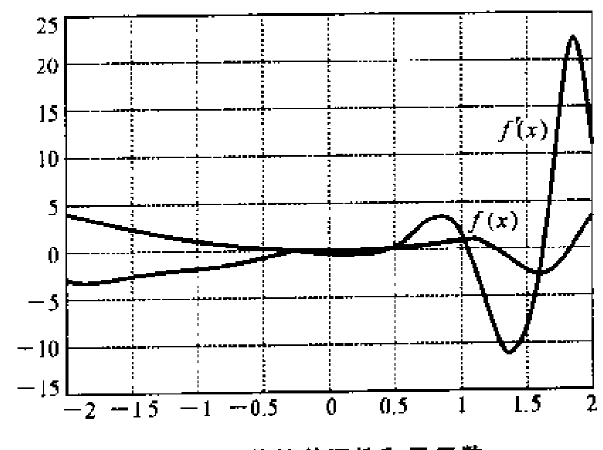


图 3-1 函数的单调性和导函数

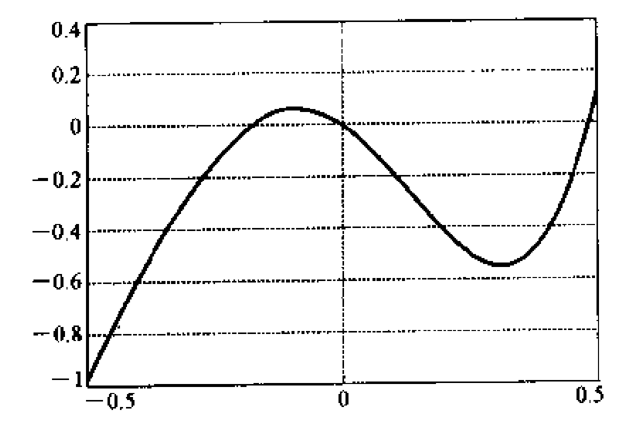


图 3-2 极值点的发现

可以明显看见，在−0.2 和 0 附近 $f'(x)$ 还有两个零点，且从 $f'(x)$ 的变化知道前者为 $f(x)$ 极小值点，后者为 $f(x)$ 极大值点（图 3-2）。

怎样算出极值点确切位置呢？当然可用 fmin 或 fmins。

```
» fmins(fun, 0.5)
```

得到一个极小值点为 0.4798。

```
» nfun = '-x*x*cos(x*x+3*x-4)'; fmins(nfun, 1.5)
```

得到极大值点为 1.0689，另一方法是通过计算 $f'(x)$ 的零点得到。

```
»fzero(dfun, 0.5), fzero(dfun, 1)
```

读者可自己求出其他三个极值点的位置。

例 5 （发现中值定理）对于函数 $f(x) = x + \cos(x)$，在 $[0, \pi/2]$ 上验证微分中值定理。

解：根据微分中值定理，在 $[0, \pi/2]$ 存在 ξ 使

$$f(\pi/2) - f(0) = f'(\xi)(\pi/2 - 0)$$

即 $f'(\xi) - (\pi/2 - 1)/(\pi/2) = 0$

几何上，必有一点的切线与两端点的连线平行。我们先作图目测一下 ξ 的位置，看得出在 0.75 左右（图 3-3）。

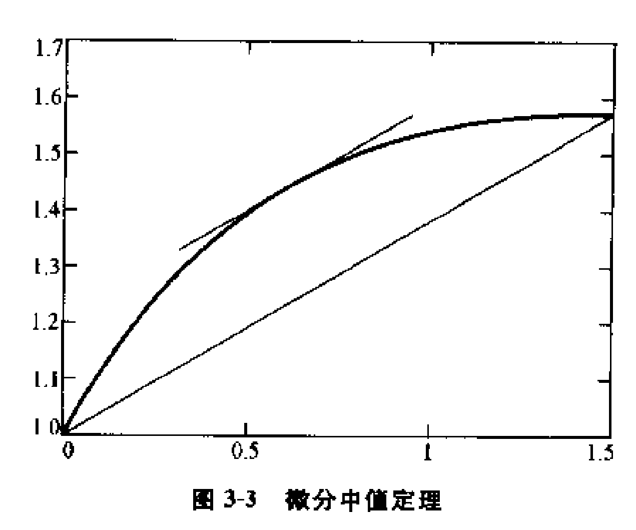


图 3-3 微分中值定理

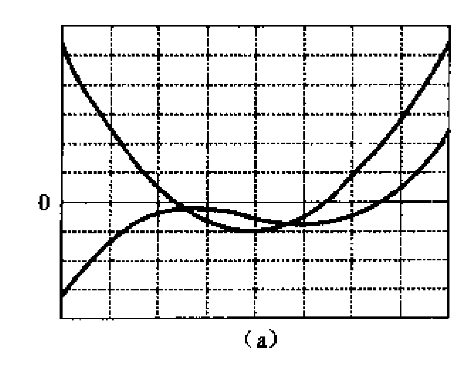
```
» close; fplot('x+cos(x)', [0, pi/2]); hold on;
» plot([0, pi/2], [1, pi/2], 'r'); %目测一下ξ的位置
```

由 $f'(x) = 1 - \sin x$，用下列命令求准确位置

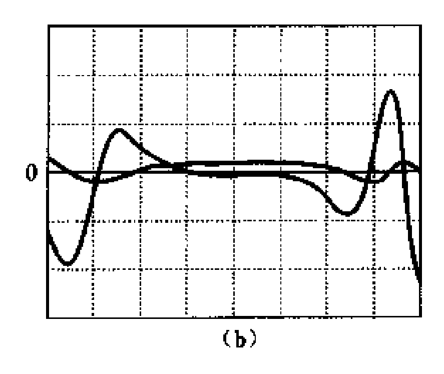
```
» ksai = fzero('1 - sin(x) - (pi/2 - 1)/pi*2', 0.75)
ksai=
  0.6901
```

§3.6 实 验 习 题

1 下列图 3-4 中各有两条曲线，辨认哪条是 $f(x)$，哪条是 $f'(x)$？为什么？



(a)



(b)

图 3-4 习题 1 图

2 （单调性）考虑函数

$$f(x) = 3x^2\sin(x^3),\ -2 < x < 2$$

(1) 作出图形,并说出大致单调区间;

(2) 使用 diff 求 $f'(x)$,并求 $f(x)$确切的单调区间。

3 (光滑性)对于

$$f(x) = \begin{cases} x^2, & x > 0 \\ x, & x \leqslant 0 \end{cases},\ g(x) = \begin{cases} x^2, & x > 0 \\ x^3, & x \leqslant 0 \end{cases}$$

(1) 作图观察它们是否连续,是否光滑?

(2) 求它们在 $x = 0$ 的左导数和右导数。

4 编制一个求二元函数偏导数的 M 函数。

5 (最优化)对于下列函数完成下列工作,并写出总结报告,评论极值与导数的关系。

(i) 作出图形,观察所有的局部极大、局部极小和全局最大、全局最小值点的粗略位置;

(ii) 求 $f'(x)$所有零点(即 $f(x)$的驻点);

(iii) 求出驻点处 $f(x)$的二阶导数值;

(iv) 用 fmin 求各极值点的确切位置;

(v) 局部极值点与 $f'(x)$, $f''(x)$有何关系?

(1) $f(x) = x^2\sin(x^2 - x - 2),\ x \in [-2, 2]$

(2) $f(x) = 3x^5 - 20x^3 + 10,\ x \in [-3, 3]$

(3) $f(x) = |x^3 - x^2 - x - 2|,\ x \in [0, 3]$

6 (通道中的细杆)要运送一根细杆子通过由宽 5m 和宽 10m 的通道垂直交叉口,在运送过程中必须保持杆子是水平的(如图 3-5),问这根细杆至多可有多长? 又通道为圆柱形的且细杆不必保持水平,细杆至多可有多长?

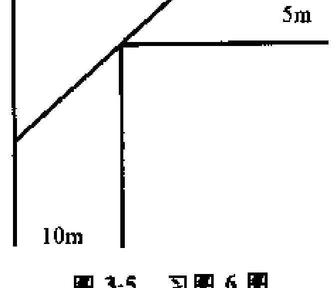


图 3-5 习图 6 图

7 (Taylor 展开)求下列函数的 Taylor 展开式 ($n = 8$)

$$\exp(x),\ \ln(1+x),\ \sin(x),\ \mathrm{tg}(x)$$



8 考虑函数

$$f(x, y) = y^3/9 + 3x^2y + 9x^2 + y^2 + xy + 9$$

(1) 作出 $f(x, y)$在 $-2 < x < 1,\ -7 < y < 1$ 的图,观察极值点的位置;

(2) 求使 $\nabla f = 0$ 的四点;

(3) 用 Hesse 矩阵判断它们是局部极大、局部极小、还是鞍点;

(4) 用 fmins 求极值点。

9 一条长凳被牢牢固定在地上,凳面水平。考虑若干块砖在长凳一端叠成阶梯状而尽量向外延伸。一块砖放在长凳右端极端位置是砖的一半在外,但第二块砖若仍放一半(如图 3-6)必会倒下。应如何放置这两块砖。n 块呢?

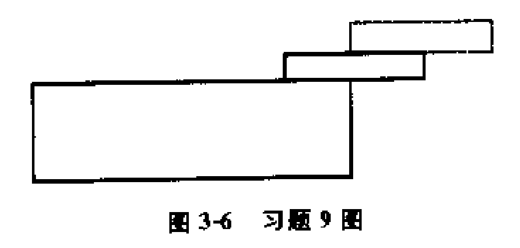


图 3-6 习题 9 图

10 (电视机价格)由于市场竞争的影响,电视

机售价 p 越高，销售量 x 就会越低，

$$x = Me^{-ap}, \qquad M, a > 0$$

其中 M 为最大需求量，a 为价格系数。另一方面，销售量越大，每台电视机成本 c 就会越低，

$$c = c_0 - k\ln x \qquad c_0, k > 0, x \gg 1$$

其中 c_0 是只生产一台电视机时的成本，k 为规模系数。应如何确定电视机售价才能获得最大利润？

11 （扩大生产的订货）在引例的订货问题中，若预计汽车生产在未来一段时间按下列函数增长

$$y = ae^{bt}$$

其中 a，b 为常数。修正经济批量订货公式。

§3.7 补充知识：计算的局局性

1. 计算的局限性

在数值计算（如迭代法）中，问题的解 x 往往是计算过程值 $x_0, x_1, \cdots, x_k, \cdots$ 的极限，但由于不可能在计算上实现这一无限序列过程，必然计算至某步 x_k 停止，$|x - x_k|$ 为解的误差。在实际计算中，通常采用下列停止准则：对于给定的精度要求 ε，当 $|x_k - x_{k-1}| < \varepsilon$，则将 x_k 作为满足精度的解。这一做法是否可靠？下列 M 函数 lmt.m 是按这一做法验证极限收敛性的数值方法，使用格式为

```
[x, n] = lmt('sequence', epsilon, maxn)
```

其中 sequence 是用 M 函数表达的序列，epsilon 是精度要求（缺省值 1e－4），maxn 是序列最大长度（缺省值 1e+6），若在 maxn 内满足精度，返回极限值 x 和下标 n，并作图；否则，认为不收敛。

```
%M 函数 lmt.m
function [x, n] = lmt(sequence, epsilon, maxn)
if nargin<3, maxn = 1e + 6; end
if nargin<2, epsilon = 1e - 4; end
n = 0; x = inf; close;
while n<maxn,
    n = n + 1; seq = feval(sequence, n);
    if abs(seq - x)<epsilon, break; end
    x = seq;
end
if n<maxn,
    plot(1:n, feval(sequence, 1:n),'.r');
    title(['LMT = ',num2str(x),', N = ', int2str(n)]);
```

```
else,
    warning(['Nonconvergence in ', int2str(maxn),' steps']);
    x = nan; n = nan;
end
```

现在我们来考虑下列几个简单的序列

$$\{(-1)^n/n\},\ \{(-1)^n/n^2\},\ \{(-1)^n/n^3\}$$

显然它们都趋于 0,对于精度 1e−4,理论上相应 n 的大小分别为 10000, 100, 22。若用上述 lmt 验证可以发现基本能反映收敛精度,但对于

$$\{1/n\},\ \{1/n^2\},\ \{1/n^3\}$$

会有较大差异。而对于

$$\{1/n^{0.01}\} \text{ 和 } \{(-1)^{[n/4]}\}$$

甚至得出错误结论(见习题 12)。分析一下为什么?

注:严格来说,上述验证极限的方法并不严密。根本问题在于数值计算是一个有限步计算,它无法解决无限步的极限问题,任何数值收敛都冒着假收敛的危险(见实验四的习题 6),这是科学计算的一个普遍缺点。收敛的根本保证只能靠理论证明(或者说人的抽象思维能力)来解决。

2. 黄金分割法

理论上,光滑函数的极值点可通过对其导函数的求根得到,但从计算的角度并不合算。首先要解析地求出导函数,求根本身也是不容易,况且求得的根并不保证总是极值点。求函数极值的实用算法是迭代搜索法,如最速下降法、Powell 算法、单纯形搜索法等。实际上人们常用这类算法求非线性方程(组)的根(如 MATLAB 函数 fsolve)。

这里介绍一元函数极值的黄金分割法,它是许多最优化搜索法的基础,其基本思想与解方程的二分法搜近。如图 3-7,设 $f(x)$ 在 $[a, b]$ 连续且有唯一的极小值点 α,设

$$a_1 = b - 0.618(b-a),\ b_1 = a + 0.618(b-a)$$

若 $f(a_1) < f(b_1)$,则 $\alpha \in [a, b_1]$;若 $f(a_1) \geq f(b_1)$,则 $\alpha \in [a_1, b]$。重复上述过程可逼近 α。M 函数 goldmin.m 求指定区间内的极小值点。使用格式

图 3-7 黄金分割法

```
x = goldmin('f_name', a, b, tol, maxsearch)
```

其中 f_name 字符串为一元优化函数 $f(x)$(用 M 函数表示),a, b 为区间下上端,tol 为精度(缺省值 1e−4),maxsearch 为最大迭代次数(缺省值 500)。

```
            %M 函数 goldmin.m
function x = goldmin(f_name, a, b, tol, maxsearcb)
if nargin<5, maxsearch = 500; end
if nargin<4, tol = 1e - 4; end; k = 0;
```

```
    while(k<500&b-a>tol)
        k=k+1; t=b-a;
        a1=b-0.618*t; b1=a+0.618*t;
        if feval(f_name,a1)<feval(f_name,b1),
            b=b1; else a=a1; end
    end; x=(b+a)/2;
    if k>=500, warning('Iteration exceeds the limitaton'); end
```

例如，为求 $f(x)=2x^5-x^4-4x^3+5$ 在$[-1,1]$内的极大值点，先将$-f(x)$写成M函数

```
        function y=fun(x)
        y=-2*x.^5+x.^4+4*x.^3-5;
```

再在命令窗口执行

```
>> goldmin('fun', -1, 1)
ans=
  -0.9135
```

3. 补充习题

12　考虑下列数列极限，用上述M函数lmt.m判别其收敛性和收敛速度，并分析结果。

$\{(-1)^n/n\}$，$\{(-1)^n/n^2\}$，$\{(-1)^n/n^3\}$，$\{1/n\}$，$\{1/n^2\}$，$\{1/n^3\}$，$\{1/n^{0.01}\}$和$\{(-1)^{[n/4]}\}$

13　用黄金分割法求函数

$$f(x)=3x^2\sin(x^3),\ -2<x<2$$

的极值点。

实验四　数学家的生日蛋糕　积分

本实验中我们学习数值积分方法和 MATLAB 有关积分计算的命令，加深对积分概念的理解，掌握积分在计算面积、体积等问题中的应用。补充知识介绍了变步长积分法和广义积分计算。

§4.1　引例：数学数的生日蛋糕

一个数学家即将要迎来他 90 岁生日。有很多学生要来祝寿，所以要做一个特大的蛋糕。为了纪念他提出的一项重要成果——口腔医学的悬链线模型，他的弟子要求蛋糕店老板将蛋糕边缘圆盘的半径作成下列悬链线函数

$$r = 2 - (\exp(2h) + \exp(-2h))/5, \quad 0 < h < 1(\text{单位:m})$$

由于蛋糕店从来没有做过这么大的蛋糕，蛋糕店老板必须要计算一下成本。这主要涉及两个问题的计算，一个是蛋糕的重量，由此可以确定需要多少鸡蛋和面粉；另一个是蛋糕表面积(除底面之外)，由此确定需要多少奶油。

老板没有上过大学，但是他很聪明。他这样考虑问题：对于一个圆盘形单层蛋糕，如图 4-1(a)旋转而成，若高为 H(m)，半径 r(m)，比重 k(kg/m^3)，则蛋糕的重量(kg)和表面积(m^2)

$$W = k\pi Hr^2$$

$$S = 2\pi Hr + \pi r^2$$

如果蛋糕是双层圆盘的，如图 4-1(b)旋转面成，每层高 $H/2$(m)，下层半径 r_1(m)，上层半径 r_2(m)，则蛋糕的重量和表面积

$$W = k\pi H(r_1^2 + r_2^2)/2$$

$$S = \pi H(r_1 + r_2) + \pi r_1^2$$

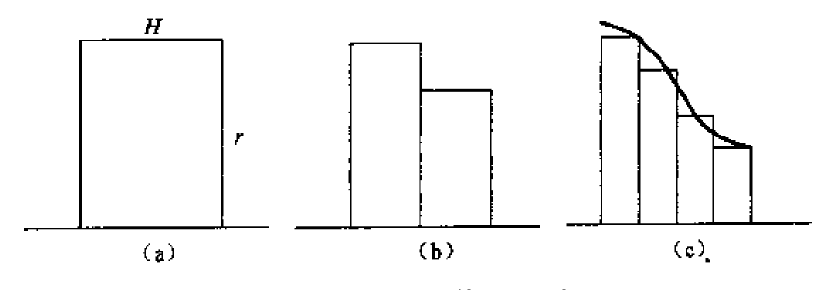


图 4-1　不同形状的生日蛋糕

虽然他只做过双层蛋糕，但可以推算，如果蛋糕是 n 层的，如图 4-1(c)旋转面成，每层高

H/n(m)，半径分别 r_1，…，r_n(m)，则蛋糕的重量和表面积

$$W = k\pi \frac{H}{n} \sum_{i=1}^{n} r_i^2$$

$$S = 2\pi \frac{H}{n} \sum_{i=1}^{n} r_i + \pi r_1^2$$

他将蛋糕假想为 4 层，然后每层的半径用区间的中点做近似，求得

$$W = 5.45k,\ S = 15.97$$

他的计算是否正确呢？事实上，若蛋糕边缘是曲线 $r = r(h)$，$0 < h < H$，各层半径近似为 $r_i = r((i-1/2)H/n)$，$i = 1, \cdots, n$，那么当 $n \to \infty$，

$$W = k\pi \frac{H}{n} \sum_{i=1}^{n} r_i^2 \to k\pi \int_0^H r^2(h)\mathrm{d}h$$

$$S = 2\pi \frac{H}{n} \sum_{i=1}^{n} r_i + \pi r_1^2 \to 2\pi \int_0^H r(h)\mathrm{d}h + \pi r(0)^2$$

§4.2 数学理论复习：积分

积分是微分的无限和，函数 $f(x)$在区间$[a, b]$上的积分定义为

$$I = \int_a^b f(x)\mathrm{d}x = \lim_{\max(\Delta x_i) \to 0} \sum_{i=1}^{n} f(\xi_i)\Delta x_i \tag{4.1}$$

其中 $a = x_0 < x_1 < \cdots < x_n = b$，$\Delta x_i = x_i - x_{i-1}$，$\xi_i \in (x_{i-1}, x_i)$，$i = 1, 2, \cdots, n$。从几何意义上说，对于$[a, b]$上非负函数 $f(x)$，积分值 I 是 $y = f(x)$ 与直线 $x = a$，$x = b$ 及 x 轴所围成的曲边梯形的面积。有界连续(或几乎处处连续)函数的积分总是存在的。

微积分基本定理(Newton-Leibniz 公式)：若在(a, b)上，$F'(x) = f(x)$，那么

$$\int_a^b f(x)\mathrm{d}x = F(b) - F(a) \tag{4.2}$$

这个公式表明导数与积分是一对互逆运算，它也提供了求积分的解析方法：为了求 $f(x)$的积分，需要找到一个函数 $F(x)$，使 $F(x)$的导函数正好是 $f(x)$，我们称 $F(x)$为 $f(x)$的原函数或不定积分。不定积分的求法常涉及许多数学技巧，常用的有换元法和分部积分法。从理论上说，可积函数的原函数总是存在的，但很多被积函数的原函数不能用初等函数表达，也就是说这些积分不能用解析方法求解，需用数值积分法解决。

在应用问题中，常常是利用微分法进行分析，面问题最终的解归结为微分的和(即积分)。一些更复杂的问题是含微分的方程，不能直接积分求解，这类问题我们放在实验五里解决。

多元函数的积分称为多重积分。二重积分定义为

$$\iint_G f(x, y)\mathrm{d}x\mathrm{d}y = \lim_{\max(\Delta x_i^2 + \Delta y_j^2) \to 0} \sum_i \sum_j f(\xi_i, \eta_j)\Delta x_i \Delta y_j$$

如图 4-2 所示，当 $f(x, y)$ 非负时，积分值几何上表示曲顶柱体的体积，二重积分的计算主要是转换为两次单积分来解决。无论是解析方法还是数值方法，如何实现这种转换，是解决问题的关键。

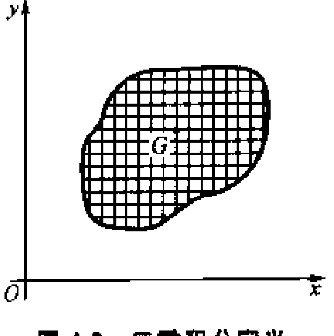


图 4-2 二重积分定义

平面曲线 $(x(t), y(t))(a < t < b)$ 的长度

$$L = \int_a^b \sqrt{x'(t)^2 + y'(t)^2}\, dt$$

空间曲线 $(x(t), y(t), z(t))(a < t < b)$ 的长度

$$L = \int_a^b \sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2}\, dt$$

曲面 $z = z(x, y)((x, y) \in G)$ 的面积

$$S = \iint_G \sqrt{1 + z_x'^2 + z_y'^2}\, dx dy$$



§4.3 数数积分:梯形法和数积分

1. 梯形法

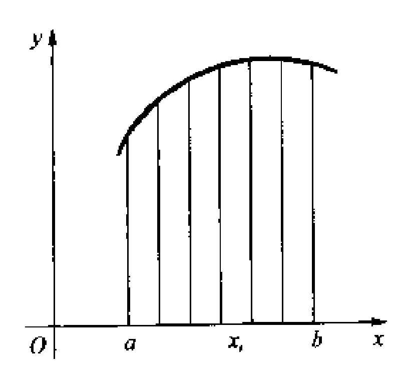


图 4-3 梯形积分法

如图 4-3 所示，设 $f(x)$ 在 $[a, b]$ 上大于 0，那么 $I = \int_a^b f(x) dx$ 就是如图 4-3 所示的曲边梯形的面积，将 $[a, b]$ 划分为若干小区间 $a = x_0 < x_1 < \cdots < x_n = b$，则

$$I = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x) dx$$

在每一小区间 $[x_{i-1}, x_i]$ 上 $f(x)$ 近似为一直线，用弦线代替，则

$$\int_{x_{i-1}}^{x_i} f(x) dx \approx \frac{x_i - x_{i-1}}{2}(f(x_{i-1}) + f(x_i))$$

从面

$$I \approx \sum_{i=1}^{n} \frac{x_i - x_{i-1}}{2}(f(x_{i-1}) + f(x_i))$$

称为梯形公式。通常将 $[a, b]$ 区间 n 等分，$h = (b - a)/n$，$x_i = a + ih$，　　(4.3)

$$I \approx T_n = h\left(\frac{1}{2}f(a) + \frac{1}{2}f(b) + \sum_{i=1}^{n-1} f(x_i)\right) \tag{4.4}$$

可以证明当 $n \to \infty$ 梯形法(4.4)式是收敛的。

2. 重积分

重积分的数值计算可通过若干次单积分的组合实现，如对于二重积分

$$I = \iint\limits_A f(x, y)\mathrm{d}x\mathrm{d}y$$

先化为二次积分

$$I = \int_a^b \mathrm{d}x \int_{c(x)}^{d(x)} f(x, y)\mathrm{d}y$$

我们利用梯形法，先将$[a, b]$区间 m 等分，$h_x = (b-a)/m$, $x_i = a + ih_x$, $i = 0, 1, \cdots, m$ 使用(4.4)式得

$$I \approx h_x\left(\frac{1}{2}(G(a)+G(b)) + \sum_{i=1}^{m-1} G(x_i)\right), \text{其中} G(x_i) = \int_{c(x_i)}^{d(x_i)} f(x_i, y)\mathrm{d}y$$

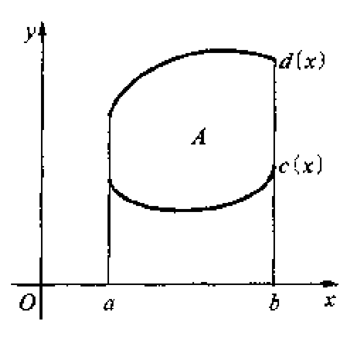


图 4-4　二重积分计算

如图 4-4，再将$[c(x_i), d(x_i)]$区间 n 等分，$h_y(i) = (d(x_i) - c(x_i))/n$, $y_{ij} = c(x_i) + jh_y(i)$, $j = 0, 1, \cdots, n$，使用(4.4)式得

$$G(x_i) \approx h_y(i)\left(\frac{1}{2}[f(x_i, c(x_i)) + f(x_i, d(x_i))] + \sum_{j=1}^{n-1} f(x_i, y_{ij})\right)$$

M 文件 dblquad2.m 给出二重积分数值计算法(其中使用了 MATLAB 梯形积分命令 trapz，见下节)。用法

```
I = dblquad2('f_name', a, b, 'c_lo','d_hi', m, n)
```

其中'f_name'为被积函数 $f(x, y)$字符串，其中 x 为标量，y 为向量，'c_lo'和'd_hi'是 y 的下限和上限函数 $c(x)$、$d(x)$，都是 x 的标量函数；a, b 分别为 x 的下限和上限；m, n 分别为 x 和 y 方向的等分数(缺省值 100)。

```
%M 函数 dblquad2.m
function I = dblquad2(f_name, a, b, c_lo, d_hi, m, n)
if nargin<7, n = 100; end
if nargin<6, m = 100; end
if m<2|n<2
    error('Number of intervals invalid');
end
mpt = m + 1; hx = (b - a)/m; x = a + (0:m)* hx;
for i = 1:mpt
    ylo = feval(c_lo, x(i)); ybi = feval(d_hi, x(i));
    hy = (ybi - ylo)/n; y(i, : ) = ylo + (0:n)* by;
    f(i, : ) = feval(f_name, x(i), y(i, : ));
    G(i) = trapz(y(i, : ), f(i, : ));
end
I = trapz(x, G);
```

例 1　$\iint\limits_{x^2+y^2\leqslant 1} \sqrt{1-x^2}\mathrm{d}x\mathrm{d}y = \int_{-1}^{1}\left(\int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} \sqrt{1-x^2}\mathrm{d}y\right)\mathrm{d}x$

```
            %M 函数 eg4_1fun.m
function z = eg4_1fun(x, y)
z = sqrt(1 - x ^ 2)*ones(size(y));
            %M 函数 eg4_1low.m
function y = eg4_1low(x)
y = - sqrt(1 - x ^ 2);
            %M 函数 eg4_1up.m
function y = eg4_1up(x)
y = sqrt(1 - x ^ 2);
```

然后在命令窗口用

```
» dblquad2('eg4_1fun', -1, 1, 'eg4_1low', 'eg4_1up')
ans =
  2.6664
```

§4.4 求积分的MATLAB命令

trapz	梯形法积分;	dblquad	矩形区域的二重积分;
quad	变步长数值积分;	int	符号积分。
quad8	高精度数值积分;		

1. 梯形积分法

trapz是最基本的数值积分方法，精度低，适用于数值函数和光滑性不好的函数。使用格式为

> z = trapz(x, y)　x是表示积分区间的离散化向量；y是与x同维数的向量，表示被积函数；z返回积分的近似值。

例2　$\int_{-1}^{1} e^{-x^2} dx$

```
»clear; x = -1:0.1:1;
»y = exp(-x.^2);
»trapz(x, y)
ans=
    1.4924
```

2. 变步长数值积分

> z = quad8('Fun', A, B, Tol)
> 　其中：Fun——表示被积函数的M函数名；
> 　　　　A, B——下限，上限；
> 　　　　Tol －精度，缺省值为1e－3。

为解例 2，先写 M 函数 eg4 _ 2fun. m

```
%M 函数 eg4 _ 2fun. m
function y = fun(x)
y = exp( - x.^ 2);
```

然后在命令窗口用

```
》z = quad8('eg4 _ 2fun', - 1, 1)
z =
  1.4936
```

注 1：quad 使用自适应步长 Simpson 法，quad8 使用自适应步长 8 阶 Newton-Cotes 法，我们建议用 quad8，它不但精度较高，且对假收敛(见习题 6)和假奇异积分(见习题 5)具有一定适应性，而 quad 较差。

注 2：trapz, quad, quad8 都不能用于求广义积分。此外由于数值方法的特点，对于一些假奇异积分也不能直接求解，如 $\int_{-2}^{1} x^{\frac{1}{3}}\mathrm{d}x$。因为数值方法对 $x^{\frac{1}{3}}$ 是通过 exp(ln(x)/3)计算，对 $x \leqslant 0$ 就会出现复数。这类情况在适当定义被积函数 $\left(x<0 \text{ 时用 } -(-x)^{\frac{1}{3}}\right)$ 后仍可正确求解。

3. 重积分

矩形区域二重积分

$$z = \int_{a}^{b}\mathrm{d}x\int_{c}^{d} f(x, y)\mathrm{d}y$$

dblquad 只能求矩形区域的二重积分，不如上述 M 函数 dblquad2 适用面广。

z = dblquad('Fun', a, b, c, d) 其中：Fun——表示被积函数 f 的 M 函数名。 a, b——变量 x 的下上限。c, d——变量 y 的下上限。

4. 符号积分

int(s)　符号表达式 s 的不定积分。 int(s, v)　符号表达式 s 关于变量 v 的不定积分。 int(s, a, b)　符号表达式 s 的定积分，a, b 分别为下、上限。 int(s, v, a, b)　符号表达式 s 关于变量 v 从 a 到 b 的定积分。

```
》syms x; t1 = int(exp( - x) + sin(x))
t1=
-exp(-x)-cos(x)
》t2 = int(exp( - x) + sin(x), 0, 1)
t2=
```

```
-exp(-1)-cos(1)+2
》t2 = vpa(t2)
t2=
1.0918
```

当 int 求不出符号解，会自动转求数值解。

```
》t3 = int(exp( - x ^ sin(x)), 0, 1)
Warning: Explicit integral could not be found.
>In D:\MATLAB11\toolbox\symbolic\@sym\int.m at line 58
t3=
int(exp(-x ^ sin(x)), x=0..1)      %说明无法求得解析解
》t3 = vpa(t3, 5)
t3=
  .45491        %这样仍可得近似解
```

int 也可用求解重积分和广义积分，为解例 1

```
》syms x y; iy = int(sqrt(1 - x ^ 2), y, - sqrt(1 - x ^ 2), sqrt(1 - x ^ 2));
》int(iy, x, -1, 1)
ans=
  8/3
```

若要计算 $\int_1^{\infty} e^{-x}\sin(x)dx$

```
》syms x; int(exp( - x)* sin(x), 1, inf)
ans=
  1/2* exp(-1)* cos(1)+1/2* exp(-1)* sin(1)
》vpa (ans, 5)
ans=
  .25417
```

注：int 的功能虽然很强大，但计算速度慢，数值计算中效率不好。

§4.5 实验例题

例 3 （生日蛋糕）现在让我们来做数学家的生日大蛋糕。

```
》syms h; r = 2 - (exp(2* h) + exp( - 2* h))/5;
》vpa(int(pi* r ^ 2, h, 0, 1), 5)
ans=
  5.4172
》r0 = subs(r, h, 0); vpa(int(2* pi* r, h, 0, 1) + pi* r0 ^ 2, 5)
ans=
  16.051
```

可见，蛋糕店老板的估算基本正确。事实上，他使用的是数值积分中矩形法。

例 4 （积分的定义）下列 M 函数 intdf.m 给出积分定义(4.1)的演示工具，使用格式

I = intdf('f _ name', a, b, n)

它给出 f _ name 表示的被积函数 $f(x)$ 在 $[a, b]$ 上的积分并作图，将 $[a, b]$ 区间 n 等分，ξ 全取小区间中心。

```
        %M 函数 intdf.m
function I = intdf(f _ name, a, b, n)
close; h = (b - a)/n; I = 0;
for i = 1:n
    x(1) = a + (i - 1)* h; x(2) = a + i* h;
    x(3) = x(2); x(4) = x(1); t = (x(3) + x(4))/2;
    y(3) = feval(f _ name, t); y(4) = y(3); I = I + h* y(3);
    fill(x, y, [1  0  0.5]  * i/n); hold  on;
end
fplot(f _ name, [a, b]);
title(['积分定义(n = ', int2str(n),')']);
xlabel(['积分值 = ', num2str(I)]); bold off
```

下列命令演示 $\sin(x)$ 在 $[0, 2]$ 上积分，注意积分准确值 $1-\cos(2)=1.4161$。

```
»for n = 1:10, intdf('sin', 0, 2, n), pause(5), end
```

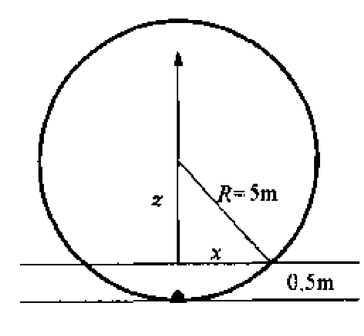


图 4-5 球形水罐

例 5 （漏水的时间）一个半径为 5m 的球形水罐充满了水，底部有一个半径 $b=0.1\text{m}$ 的小孔漏水(图 4-5)，若不考虑磨擦力，多少时间以后，水面将下降至离底部 0.5m？

解：水从小孔漏出的速度由下列能量方程决定

$$g(z+R)=u^2/2$$

这里 u 是速度，z 表示从球心测量的水面高度，$g=9.81\text{m/s}^2$ 为重力加速度。考虑在时间 dt 内水面变化 dz，漏水的体积为

$$uA\,dt=-\pi x^2 dz$$

其中 z 为高度 x 的水面的半径，小孔面积 $A=\pi b^2$。由于

$$R^2=z^2+x^2$$

从而

$$dt=-\frac{R^2-z^2}{b^2\sqrt{2g(z-R)}}dz$$

在顶部，$z=R$ 水降到 0.5m 时，$z=0.5-R$，从面

$$t=-\int_R^{0.5-R}\frac{R^2-z^2}{b^2\sqrt{2g(z+R)}}dz=\int_{0.5-R}^{R}\frac{R^2-z^2}{b^2\sqrt{2g(z+R)}}dz$$

计算程序如下，结果为 0.5144 小时。

```
        %M 脚本 eg4_5.m
clear; R=5; b=0.1; g=9.81; z1=0.5-R; z2=R;
n=100; h=(z2-z1)/n; z=z1:h:z2;
f=(R^2-z.^2)./(b^2*sqrt(2*g*(z+R)));
I=trapz(z, f)/60/60
```

§4.6 实验习题

1 (不定积分)用 int 计算下列不定积分,并用 diff 验证

$$\int \frac{e^{2y}}{e^y+2}dy,\ \int \frac{x^2}{\sqrt{a^2-x^2}}dx,\ \int \frac{dx}{x(\sqrt{\ln x+a}+\sqrt{\ln x-b})}(a\neq b)$$

2 (定积分)分别用 trapz、quad8、int 计算下列定积分

$$\int_0^1 \frac{\exp\left(-\frac{x^2}{2}\right)}{\sqrt{2\pi}}dx,\ \int_0^1 \frac{\sin(x)}{x}dx,\ \int_0^1 x^{-x}dx,\ \int_0^{2\pi}\exp(2x)\sin^2(x)dx$$

3 (椭圆的周长)用积分法计算下列椭圆的周长

$$\frac{x^2}{4}+\frac{y^2}{9}=1$$

4 (重积分)

(1) $\int_0^{2\pi}d\theta\int_0^1\sqrt{1+r^2\sin(\theta)}\,dr$

(2) $\iint\limits_D dydx$, D 为 $x^2+y^2\leqslant 2x$

5 (假奇异积分)试求下列积分

$$I=\int_{-1}^1 x^{0.2}\cos(x)dx$$

出现什么问题?分析原因,设法求出正确的解。

6 (假收敛现象)考虑积分 $I(k)=\int_0^{k\pi}|\sin(x)|dx$,

(1) 用解析方法求 $I(k)$;

(2) 试分别用 trapz、quad 和 quad8 求解 $I(4)$、$I(6)$ 和 $I(8)$,发现什么问题?

7 (Simpson 积分法)编制一个定步长 Simpson 法数值积分程序。计算公式为

$$I\approx S_n=\frac{h}{3}(f_1+4f_2+2f_3+4f_4+\cdots+2f_{n-1}+4f_n+f_{n+1})$$

其中 n 为偶数,$h=(b-a)/n$, $f_i=f(a+(i-1)h)$, $i=1,2,\cdots,n+1$。

8 (汽车)一个重 5400kg 的汽车在以速度 30m/s 行验时突然熄火,设滑行方程为

$$5400v\frac{dv}{dx}=-8.276v^2-2000$$

x 为滑行距离，v 为车速，计算要滑行多长距离后，速度可降至 15m/s。

9 （排洪量）某河床的横断面如图 4-6 所示，为了计算最大的排洪量，需要计算它的断面积，试根据图示测量数据（单位：m）用梯形法计算其断面积。

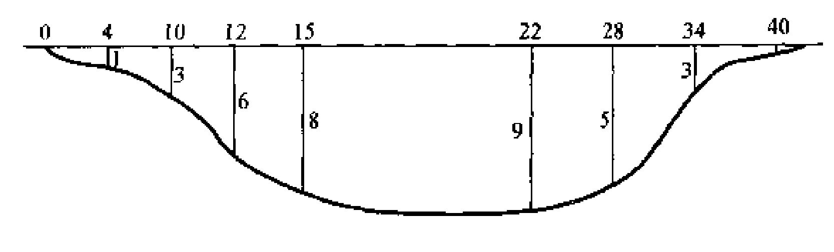
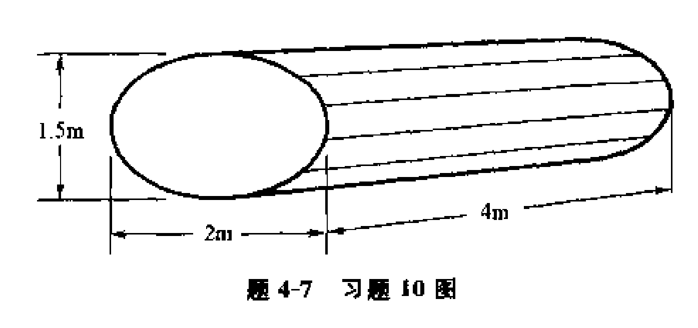


图 4-6 习题 9 图



题 4-7 习题 10 图

10 （水箱压力）洒水车上水箱是一个横放的椭圆柱体，尺寸如图 4-7 所示，当水箱盛满水时，计算两个端面所受的压力。

11 （停产时间）某公司投资 2000 万元建成一条生产线。投产后，在时刻 t 的追加成本和追加收益分别为 $G(t)=5+2t^{2/3}$（百万元/年），$H(t)=17-t^{2/3}$（百万元/年）。试确定该生产线在何时停产可获最大利润？最大利润是多少？

12 （教堂顶部曲面面积）某个阿拉伯国家有一座著名的伊斯兰教堂，它以中央大厅的金色巨大拱形圆顶名震遐迩。因年久失修，国王下令将教堂顶部重新贴金箔装饰。据档案记载，大厅的顶部形状为半球面，其半径为 30m。考虑到可能的损耗和其他技术因素，实际用量将会比教堂顶部面积多 1.5%。据此，国王的财政大臣拨出了可制造 5800m^2 有规定厚度金箔的黄金。建筑商人哈桑略退数学，他计算了一下，觉得黄金会有盈余。于是，他以较低的承包价得到了这项装饰工程。但在施工前的测量中，工程师发现教堂顶部实际上并非是一个精确的半球面面是半椭球面，其半立轴恰是 30m，而半长轴和半短轴分别是 30.6m 和 29.6m。这一来哈桑犯了愁，他担心黄金是否还有盈余？甚至可能短缺。最后的结果究竟如何呢？

§4.7 补充知识：变步长积分法

1. 变步长积分法

在以上介绍的梯形法中，划分是给定的。在实际应用中，等分数 n 柱往是难以确定的。下面介绍变步长梯形法，其思路为，对于给定的精度，从 $n=1$ 开始，等分数逐次加倍 $n=2, 4, 8, \cdots$，直至 $|T_{2n}-T_n|<\varepsilon$ 为止，一般来说，这样的做法可能会浪费计算量，幸运的是，T_{2n} 与 T_n 有如下递推关系

$$T_{2n} = \frac{1}{2}T_n + \frac{h}{2}\sum_{i=1}^{n} f\left(a + \frac{1}{2}ih\right)$$

其中 $h = (b-a)/n$。M 函数 trapz _ v. m 是变步长梯形法，其使用格式为

```
[t, n] = trapz _ v('fun', a, b, tol)
```

tol 是精度（缺省值 1e−4）。对于发散或过慢收敛，设置等分数上限 2^{30}，为了防止假收敛（见习题 6）设置等分数下限 2^5。t 返回积分值，n 返回等分数。

```
            %M 函数 trapz _ v. m
function [t, n] = trapz _ v(f _ name, a, b, tol)
if nargin<4, tol = 1e - 4; end
fa = feval(f _ name, a); fb = feval(f _ name, b);
b = b - a; t = h* (fa + fb)/2; t0 = t + 2* tol;
minpt = 2 ^ 5; maxpt = 2 ^ 30; n = 1;
while(abs(t - t0)>tol & n<maxpt)|n<minpt,
    t0 = t; x = (a + h/2):h:b; f = feval(f _ name, x);
    t = t0 + h* sum(f); t = t/2; h = b/2; n = 2* n;
end
if n>= maxpt, warning('Iteration exceeds limit'); end
```

2. 广义积分数值计算

广义积分数值求解是一个较困难的问题，通常数值积分方法都不适用。

（1）无界广义积分，如

$$I = \int_{-\infty}^{\infty} \exp(\sin x - x^2/100)\mathrm{d}x$$

若先写一个 M 函数 fun. m

```
function y = fun(x)
y = exp(sin(x) - x.^ 2/100);
```

再用命令

```
»quad8('fun', - inf, inf);
```

结果为 NaN。为能正确求解，考虑到

$$I = \lim_{N\to\infty}\int_{-N}^{N} \exp(\sin x - x^2/100)\mathrm{d}x$$

但问题是 N 取多大？比如用

```
»quad8('fun', - 1e + 10, 1e + 10)
```

结果得一个明显错误的大数。正确方法是，先取一适当大的 N（如 10），计算一个 I 值；然后将 N 以一适当倍数（如 2）增加计算出新的 I 值，直至前后两次差异小于给定精度为止。

```
»clear; n = 10; r = 2; e = 1e - 4; t0 = inf; t1 = quad8('fun', - n, n);
»while abs(t0 - t1)>e, t0 = t1; n = n* r; t1 = quad8('fun', - n, n), end
```

计算得 22.4404。

(2) 奇点积分，如

$$I=\int_0^1 \frac{dx}{\sqrt{x}(\exp(x)+1)}$$

先写一个 M 函数 fun.m

```
function y = fun(x)
y = 1./(sqrt(x).*(exp(x)+1));
```

再用命令

》quad8('fun', 0, 1)

得结果 Inf(显然不对，因为积分收敛)

》quad8('fun', 1e-5, 1), quad8('fun', 1e-10, 1)

得 0.8362 和 1.6861，系统警告结论不可靠，哪个正确？

较可靠的方法是使用指数变换

$$x=\frac{1+\tanh(z)}{2}=\frac{\exp(z)}{\exp(z)+\exp(-z)}$$

化为无界积分

$$I=\int_{-\infty}^{\infty} dz\Big/\left(\sqrt{\frac{1+\tanh(z)}{2}}\left(\exp\left(\frac{1+\tanh(z)}{2}\right)+1\right)(2\cosh^2(z))\right)$$

然后用无界积分的方法得 0.8389。

注：Symbolic 命令 int 可求广义积分。

```
》syms x;y = int(exp(sin(x) - x^2/100), -inf, inf); vpa(y, 5)
ans=
  22.440
》syms x; y = int(1/sqrt(x)/(exp(x)+1), 0, 1); vpa(y, 5)
ans=
 .83896
```

3. 补充习题

13 用变步长梯形法解第 6 题，使精度达到 1e−5。

14 计算广义积分

$$\int_{-\infty}^{\infty}\frac{\exp(-x^2)}{1+x^2}dx,\ \int_0^1\frac{\tan(x)}{x^{0.7}}dx,\ \int_0^1\frac{\exp(x)}{\sqrt{1-x^2}}dx$$

实验五　导弹系统的改进　微分方程

本实验中我们学习常微分方程(组)数值求解 Euler 法和 MATLAB 有关求解常微分方程(组)的命令。介绍导弹系统改进和产品销售量等几个实际问题的微分方程建模及求解方法。补充知识介绍了四阶 Runge-Kutta 法、数值稳定性和微分方程的稳定性理论等。

§5.1　引例:导弹系统的改进

海军方面要求改进现有的舰对舰导弹系统。目前的电子系统能迅速测出敌舰的种类、位置以及敌舰行驶速度和方向,且导弹自动制导系统能保证在发射后任一时刻都能对准目标。根据情报,这种敌舰能在我军舰发射导弹后 T 小时作出反应并摧毁导弹。现在要求改进电子导弹系统使之能自动计算出敌舰是否在有效打击范围之内。

如图 5-1 所示,设我舰发射导弹时位置在坐标原点,敌舰在 x 轴正向 d(km)处,其行驶速度为 a(km/h),方向与 x 轴夹角为 θ,导弹水平飞行线速度 b(km/h)。问题的关键是求出导弹击中敌舰的时间。

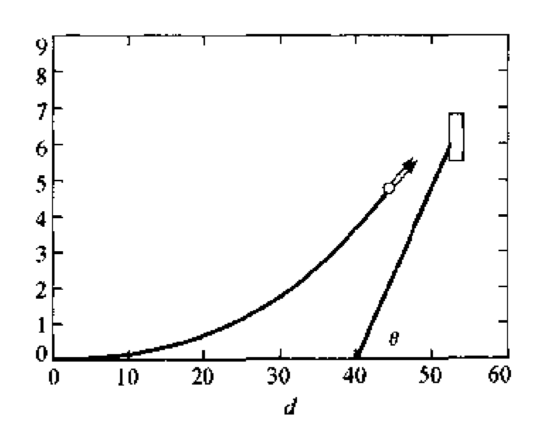


图 5-1　导弹攻击示意图

设 t 时刻导弹位置为$(x(t),\ y(t))$,那么

$$\sqrt{\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)^2+\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2}=b \qquad (5.1)$$

易知 t 时刻敌舰位置为 $(d+at\cos\theta,\ at\sin\theta)$,为了保持对准目标,导弹轨迹切线方向应为

$$\frac{\mathrm{d}y}{\mathrm{d}x}=\frac{at\sin\theta-y(t)}{d+at\cos\theta-x(t)} \qquad (5.2)$$

由(5.1)式、(5.2)式得下列微分方程

$$\frac{\mathrm{d}x}{\mathrm{d}t}=\frac{b}{\sqrt{1+\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2}}=\frac{b}{\sqrt{1+\left(\frac{at\sin\theta-y(t)}{d+at\cos\theta-x(t)}\right)^2}} \qquad (5.3)$$

$$\frac{\mathrm{d}y}{\mathrm{d}t}=\frac{b}{\sqrt{1+\left(\frac{\mathrm{d}x}{\mathrm{d}y}\right)^2}}=\frac{b}{\sqrt{1+\left(\frac{d+at\cos\theta-x(t)}{at\sin\theta-y(t)}\right)^2}} \qquad (5.4)$$

初始条件 $x(0)=0,\ y(0)=0$。对于给定的 $a,\ b,\ d,\ \theta$ 进行计算。当 $x(t)$满足

$$x(t)>d+at\cos\theta \qquad (5.5)$$

则认为已击中目标。如果 $t < T$，则敌舰在打击范围内，可以发射。

§5.2 数学理论复习：常数分方程

1. 微分方程的程论

未知的函数以及它的某些阶的导数连同自变量都由一已知方程联系在一起的方程称为**微分方程**。如果未知函数是一元函数，称为**常微分方程**。如果未知函数是多元函数，称为**偏微分方程**。联系一些未知函数的一组微分方程称为**微分方程组**。微分方程中出现的未知函数的导数的最高阶数称为**微分方程的阶**。若方程中未知函数及其各阶导数都是一次的，称为**线性常微分方程**，一舰表示为

$$y^{(n)} + a_1(t)y^{(n-1)} + \cdots + a_{n-1}(t)y' + a_n(t)y = b(t) \tag{5.6}$$

若(5.6)式系数 $a_i(t)(i = 1, 2, \cdots, n)$ 均与 t 无关，称之为**常系数**(或**定常**、**自治**、**时不变**)的。

2. 初等积分法

有些微分方程可直接通过积分求解。例如，一阶常系数线性常微分方程

$$y' = ay + b \quad (a \neq 0) \tag{5.7}$$

可化为

$$\frac{\mathrm{d}y}{ay+b} = \mathrm{d}t \tag{5.8}$$

两边积分可得通解 $y(t)$ 为

$$y(t) = C\exp(at) - a^{-1}b \tag{5.9}$$

其中 C 为任意常数。有些常微分方程可用一些技巧(如分离变量法、积分因子法、常数变易法、降阶法等)化为可积分的方程而求得显式解。

3. 常系数线性微分方程

线性常微分方程的解满足叠加性原理，从而它的求解可归结为求一个特解和相应齐次微分方程的解。一阶变系数线性常微分方程总可用这一思路求得显式解。高阶线性常系数微分方程可用特征根法求得相应齐次微分方程的基本解，再用常数变易法求特解。

例1　求 $x'' + 0.2x' + 3.92x = 0$ 的通解。

解：特征方程为

$$\lambda^2 + 0.2\lambda + 3.92 = 0$$

```
》roots([1 0.2 3.92])
```

求得共轭复根 $-0.1 \pm 1.9774\mathrm{i}$，从而通解为

$$x(t) = A\mathrm{e}^{-0.1t}\cos(1.9774t) + B\mathrm{e}^{-0.1t}\sin(1.9774t)$$

其中 A, B 为任意常数。

一阶常微分方程组与高阶常微分方程可以互化，已给一个 n 阶方程

$$y^{(n)} = f(t, y, y', \cdots, y^{(n-1)}) \tag{5.10}$$

设 $y_1 = y$，$y_2 = y'$，…，$y_n = y^{(n-1)}$，(5.10)化为一阶方程组

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ \cdots \\ y_{n-1}' = y_n \\ y_n' = f(t, y_1, y_2, \cdots, y_n) \end{cases} \tag{5.11}$$

反过来，在许多情况下，一阶微分方程组也可化为高阶方程。所以一阶常微分方程组与高阶常微分方程的理论与方法在很多方面是相通的。一阶常系数线性微分方程组也可用特征根法求解。

§5.3 微分方程微值解法:Euler 法

除常系数线性微分方程可用特征根法求解，少数特殊方程可用初等积分法求解外，大部分微分方程无显式解，应用中主要依靠数值解法。考虑一阶常微分方程组初值问题

$$\begin{cases} y'(t) = f(t, y(t)),\ t_0 < t < t_f \\ y(t_0) = y_0 \end{cases} \tag{5.12}$$

其中 $y = (y_1, y_2, \cdots, y_m)'$，$f = (f_1, f_2, \cdots, f_m)'$，$y_0 = (y_{10}, y_{20}, \cdots, y_{m0})'$。**所谓数值解法**，就是寻求解 $y(t)$ 在一系列离散节点 $t_0 < t_1 < \cdots < t_n \leqslant t_f$ 上的近似值 $y_k(k = 0, 1, \cdots, n)$。称 $h_k = t_{k+1} - t_k$ 为步长，通常取为常量 h。最简单的数值解法是 Euler 法。

Euler 法的思路极其简单：在节点处用差商近似代替导数

$$y'(t_k) \approx \frac{y(t_{k+1}) - y(t_k)}{h}$$

这样导出计算公式(称为 **Euler 格式**)

$$y_{k+1} = y_k + hf(t_k, y_k),\ k = 0, 1, 2, \cdots \tag{5.13}$$

它能求解各种形式的微分方程。Euler 法也称折线法。

M 函数 euler.m 给出定步长 Euler 法计算程序，其使用格式为

[tout, yout] = euler('ypfun', tspan, y0, h)

这里字符串 ypfun 是用以表示 $f(t, y)$ 的 M 文件名，tspan = $[t_0, t_f]$ 表示自变量初值 t_0 和终值 t_f，y0 表示初值向量 y_0，h 是步长。输出列向量 tout 表示节点 $(t_0, t_1, \cdots, t_n)'$，输出矩阵 yout 表示数值解，每一列对应 y 的一个分量。

```
        %M 函数 euler.m
function [tout, yout] = euler(ypfun, tspan, y0, h)
t = tspan(1):h:tspan(2); y(:, 1) = y0(:);
for i = 1:length(t) - 1,
    y(:, i + 1) = y(:, i) + h* feval(ypfun, t(i), y(:, i));
```

```
end
tout = t'; yout = y';
```

例2 解方程

$$y' = y - 2t/y,\ y(0) = 1,\ 0 < t < 1 \tag{5.14}$$

其准确解为 $y = \sqrt{1+2t}$。先写 M 函数 eg5_2fun.m

```
%M 函数 eg5_2fun.m
function f = fun(t, y)
f = y - 2*t./y;
f = f(:);             %保证 f 为一个列向量
```

再在命令窗口用

```
»clear; close; t = 0:0.1:1;
»y = sqrt(1 + 2*t); plot(t, y); hold on;
»[t, y] = euler('eg5_2fun', [0, 1], 1, 0.1); plot(t, y, 'r.');
»legend('准确解','Euler 法');
```

Euler 法只有一阶精度，改进方法有二阶 Runge-Kutta 法、四阶 Runge-Kutta 法、五阶 Runge-Kutta-Felhberg 法和线性多步法等。根据(5.10)式到(5.11)式的转化，这些方法可用于解高阶常微分方程(组)初值问题。边值问题采用不同方法，如差分法、有限元法等(参见文献[9])。数值算法的主要缺点是它缺乏物理解释。

§5.4 解微分方微的 MATLAB 命令

ode23	二、三阶 Runge-Kutta 法；	ode15s	刚性方程组解法；
ode45	四、五阶 Runge-Kutta 法；	dsolve	符号解析解。

1. 数值解

> [tout, yout] = ode45('yprime', tspan, y0)
> 这里字符串 yprime 是用以表示 $f(t, y)$ 的 M 文件名，tspan = $[t_0, t_f]$ 表示自变量初值 t_0 和终值 t_f，y0 表示初值向量 y_0。输出列向量 tout 表示节点 $(t_0, t_1, \cdots, t_n)^T$，输出矩阵 yout 表示数值解，每一列对应 y 的一个分量。若无输出参数，则自动作出图形。

ode45 是最常用的求解微分方程的命令。它采用变步长四阶 Runge-Kutta 法和五阶 Runge-Kutta-Felhberg 法，对于刚性方程组不宜采用。ode23 与 ode45 类似，只是精度低一些。为求解例2

```
»[t, y] = ode45('eg5_2fun', [0, 1], 1); plot(t, y, 'g:');
»hold off;
```

2. 符号微分方程解析解

s = dsolve('方程 1','方程 2'…,'初始条件 1','初始条件 2',…,'自变量')。
均用字符串方式表示,自变量缺省值为 t。
导数用 D 表示,2 阶导数用 D2 表示,以此类推。
s 返回解析解。方程组情形,s 为一个符号结构。

例 3 (1) 求 $y' = ay + b$ 的通解;

(2) 求解例 2;

(3) 高阶方程 $y'' = \cos(2x) - y$, $y(0) = 1$, $y'(0) = 0$

(4) 方程组 $f' = f + g$, $g' = -f + g$, $f(0) = 1$, $g(0) = 2$

```
》s = dsolve('Dy = a* y + b')
s=
(-b+exp(a*t)*C1*a)/a
》dsolve('Dy = y - 2* t/y', 'y(0) = 1')
ans=
(2*t+1)^(1/2)
》s = dsolve('D2y = cos(2* x) - y', 'y(0) = 1', 'Dy(0) = 0', 'x'), simplify(s)
ans =
-2/3*cos(x)^2 + 1/3 + 4/3*cos(x)
》S = dsolve('Df = f + g', 'Dg = -f + g', 'f(0) = 1', 'g(0) = 2');
》S.f, S.g          %S 是一个结构
ans=
exp(t)*cos(t)+2*exp(t)*sin(t)
ans=
-exp(t)*sin(t)+2*exp(t)*cos(t)
```

3. 刚性方程组解法

刚性方程组解法 ode15s 使用格式同 ode45。

例 4

$$\begin{cases} y_1' = -0.01y_1 - 99.99y_2, & y_1(0) = 2 \\ y_2' = \qquad\qquad -100y_2, & y_2(0) = 1 \end{cases} \tag{5.15}$$

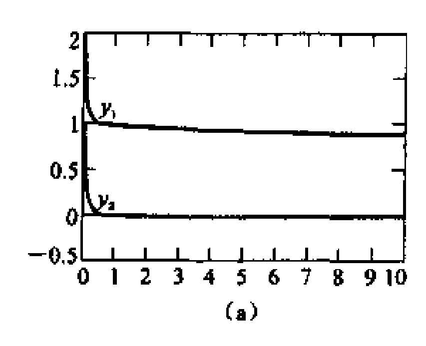
```
%M 函数 eg5_4fun.m
function f = fun(t, y)
f = [-0.01 -99.99; 0 -100]*y;
f = f(:);
```

若用 ode45 来解

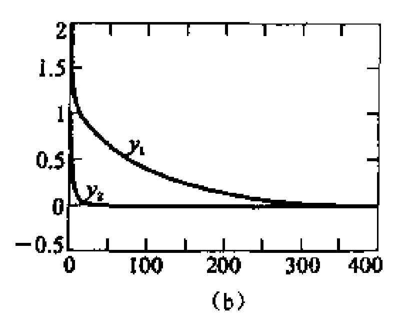
```
》clear; close; [t, y] = ode45('eg5_4fun', [0, 10], [2, 1]');
》plot(t, y); text(1, 1.1, 'y1'); text(1, 0.1, 'y2');
```

给人的感觉似乎是,y_1 始终大于 0.5(图 5-2(a))。使用 dsolve 可知解析解 $y_1 = \exp(-0.01t) + \exp(-100t)$, $y_2 = \exp(-100t)$。所以 $t \to \infty$,两个分量均→0。但 y_2 下降

极快，$y_2(0.1) < 0.0001$；而 y_1 下降很慢，$y_1(400) \approx 0.0183$（图 5-2(b)）。若用



(a)



(b)

图 5-2 刚性方程组

(a) $0 < t < 10$；(b) $0 < t < 400$

```
»[t, y]=ode45('eg5_4fun', 0, 400, [2, 1]');
»tstep=length(t), minh=min(diff(t)), maxh=max(diff(t))
tstep=
  48261
minh=
  5.0238e-004
maxh=
  0.0102
```

可见计算太慢，t 需 48261 步才能达到 400。一方面，由于 y_2 下降太快，为了保证数值稳定性（见补充知识），步长 h 需足够小；另一方面，由于 y_1 下降太慢，为了反映解的完整性，时间区间需足够长，这就造成计算量太大。这类方很组称为刚性方稳组成病态方程组。ode45 不适用于病态方程组，下面我们用 ode15s 法求解。

```
»[t, y]=ode15s('eg5_4fun', [0, 400], [2, 1]');
»plot(t, y); text(100, 0.5, 'y1'); text(1, 0.1, 'y2');
»length(t), min(diff(t)), max(diff(t))
```

可见只需 92 步，最大步长为 32。速度快了约 500 倍。

§5.5 实 验 例 题

例 5 （*导弹系统改进*）在引例导弹系统中设 $a=90\text{km/h}$，$b=450\text{km/h}$，$T=0.1\text{h}$。现要求 d，θ 的有效范围。

解：有两个极端情形容易算出。若 $\theta=0$，即敌舰正好背向行驶，即 x 轴正向。那么导弹直线飞行，击中时间

$$t = d/(b-a) < T$$

得 $d = T(b-a) = 36\text{km}$。若 $\theta=\pi$，即迎而驶来，类似有 $d = T(a+b) = 54\text{km}$。一般地，应有 $36 < d < 54$。

(1) *在线算法* 对于测定的 d 和 θ，可用(5.3)式、(5.4)式计算出 t。比如 $d = 50$，

$\theta = \frac{\pi}{2}$，写 M 函数 eg5 _ 5fun. m，为了防止分母为 0，加一个小正数 1e−8。

%M 函数 eg5 _ 5fun

```
function dy = eg5_5fun(t, y)
global a b d theta;
dydx = (abs(a* t* sin(theta) - y(2)) + 1e - 8)/...
    (abs(d + a* t* cos(theta) - y(1)) + 1e - 8);
dy(1) = b/(1 + dydx ^ 2) ^ 0.5;
dy(2) = b/(1 + dydx ^ ( - 2)) ^ 0.5; dy = dy(:);
```

在命令窗口执行

```
》clear; close;
》global a b d theta;
》a = 90; b = 450; d = 50; theta = pi/2;
》[t, y] = ode45('eg5_5fun', [0, 0.1], [0 0]);
》plot(y(:, 1), y(:, 2));
》max(y(:, 1) - d - a* t* cos(theta))
ans =
  -5.7410
```

由于在 T 小时内，(5.5)式不成立，所以敌舰不在有效打击范围，应等近一些再发射。

(2) **离线算法**　首先对于所有可能的 d 和 θ，计算击中所需时间，从而对不同 θ，得 d 的临界值。具体应用时直接查表判断。编写 M 脚本文件 eg5 _ 5. m

%M 脚本 eg5 _ 5

```
clear; close;
global a b d theta;
a = 90; b = 450; d = 50; theta = pi/2;
i = 1;
for d = 54: - 1:36
    for theta = 0:0.1:pi,
        [t, y] = ode45('eg5_5fun', [0, 0.1], [0 0]);
        if max(y(:, 1) - d - a* t* cos(theta))>0,
            range(i, :) = [d, theta]; i = i + 1; break;
        end; end; end
figure;
plot(range(:, 1), range(:, 2));
xlabel('d'); ylabel('theta');
```

运行得临界曲线。

图 5-3 中，曲线上方为打击范围。由于 $\theta = 1.57$，$d = 50$ 在曲线下方，这样即可知不在打击范围内。在线算法灵活容易调整参数和模型，但速度慢。离线算法事先计算好，实时使用查询方式，不需计算，速度极快。

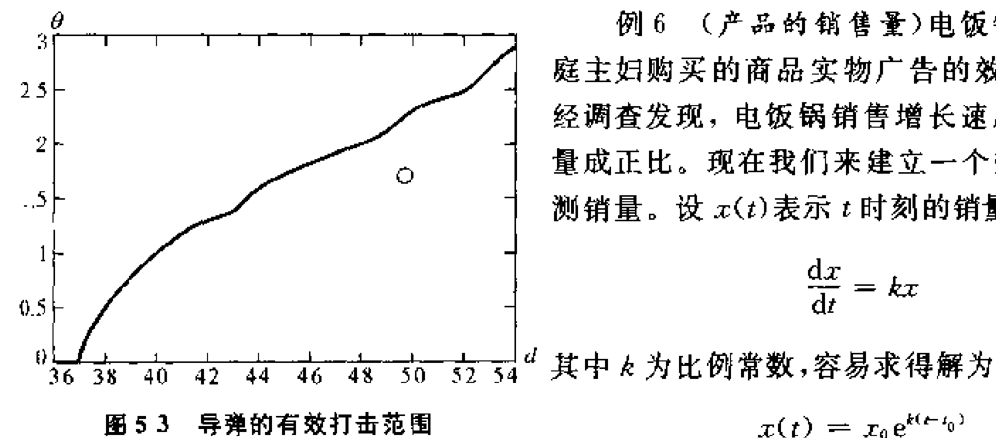


图5 3 导弹的有效打击范围

例6 (产品的销售量)电饭锅这一类的家庭主妇购买的商品实物广告的效果是很大的。经调查发现，电饭锅销售增长速度与当时的销量成正比。现在我们来建立一个数学模型以预测销量。设 $x(t)$ 表示 t 时刻的销量，那么

$$\frac{\mathrm{d}x}{\mathrm{d}t} = kx \tag{5.16}$$

其中 k 为比例常数，容易求得解为

$$x(t) = x_0 e^{k(t-t_0)} \tag{5.17}$$

这里 x_0 为初始时刻 t_0 的销量。当 $k > 0$，$x(t)$ 随 t 增长而指数状爆炸式增长，这对于销售初期可以认为是合适的。设 $t_0 = 0$ (年)，$x_0 = 1$ (万台)，$k = 0.9$ (年$^{-1}$)，可用下列命令作出6年内电饭锅销量预测图形

```
》close; fplot('exp(0.9*x)', [0, 6])
```

但这一模型是在市场容量无限的假设下取得，当 $t \to \infty$ 时，$x(t) \to \infty$，可见(5.16)式用于长期预报显然是荒唐的。为了考虑市场容量的限制，设 x_∞ 为全部需要量，那么销售速度与当时的潜在需要量 $(x_\infty - x)$ 成正比。从而得

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \lambda x(x_\infty - x) \tag{5.18}$$

其中 λ 为比例常数。可用dsolve

```
》dsolve('Dx = a*x*(x1 - x)', 'x(t0) = x0')
ans=
x1/(1 + exp(-a*x1*t)*exp(a*x1*t0)*(x1 - x0)/x0)
```

解得

$$x(t) = \frac{x_\infty}{1 + \left(\frac{x_\infty}{x_0} - 1\right)\exp(-\lambda x_\infty(t - t_0))} \tag{5.19}$$

设 $t_0 = 0$ (年)，$x_0 = 1$ (万台)，$x_\infty = 100$ (万台)，$\lambda = 0.01$ (年$^{-1}$·万台$^{-1}$)，可用下列命令作出8年内电饭锅销量预测图形

```
》hold on; fplot('100/(1 + 99*exp(-0.01*100*x))', [0, 8])
```

可见短期预报二者相近，但作为中长期预报，(5.18)式较(5.16)式合理。当然(5.18)式也有不尽合理之处，比如 x_∞ 难以确定，未考虑产品更新换代等。

§5.6 实验习题

1 用diff直接验证(5.17)式、(5.19)式分别为(5.16)式、(5.18)式的解。

2 分别用Euler法(取 $h = 0.1$) 和ode45解下列微分方程并与解析解比较

(1) $y' = x + y$, $y(0) = 1$, $0 < x < 3$

(2) $y'' - 0.01(y')^2 + 2y = \sin(t)$, $y(0) = 0$, $y'(0) = 1$, $0 < t < 5$

3　求一通过原点的曲线，它在(x, y)处的切线斜率等于$2x+y^2$。

4　试求解

$$x' = ax + b,\ x(0) = x_0$$

并分别对a, b, x_0取正负值的8种不同情况，讨论解曲线的单调性及$t\to\infty$时的行为。用MATLAB画出解曲线图形。将它们合理分类。

5　（温度过程）夏天把开有空调的室内一支读数为20℃的温度计放到户外，10分钟后读数为25.2℃，再过10分钟后读数为28.32℃。建立一个较合理的模型来推算户外温度。

6　（广告效应）某公司生产一种耐用消费品，市场占有率为5%时开始做广告，一段时间的市场跟踪调查后，该公司发现：单位时间内购买人口百分比的相对增长率与当时还没有买的百分比成正比，且估得此比例系数为0.5。

(1) 建立该问题的数学模型，并求其数值解；

(2) 厂家问：要做多少时间广告，可使市场购买率达到80%？

7　（肿瘤生长）肿瘤大小V生长的速率与V的a次方成正比，其中a为形状参数，$0\leqslant a\leqslant 1$；而其比例系数K随时间减小，减小速率又与当时的K值成正比，比例系数为环境参数b。设某肿瘤参数$a=1$，$b=0.1$，K的初始值为2，V的初始值为1。问

(1) 此肿瘤生长不会超过多大？

(2) 过多长时间肿瘤大小翻一倍？

(3) 何时肿瘤生长速率由递增转为递减？

(4) 若参数$a=2/3$呢？

8　（RLC电路）在RLC含温串联电路中，电动势为E的电温对电容器C充电。已知电阻$R=1000\Omega$，电感$L=0.1\text{Hn}$，$C=0.2\mu\text{F}$，$E=20\text{V}$，试求合上开关K后的电压$u_C(t)$。

9　（生态系统的振荡现象）第一次世界大战中，因为战争很少捕鱼，按理战后应能捕到很多的鱼才是。可是大战后，在地中海却捕不到鲨鱼，因而渔民大惑不解。

令x_1为鱼饵的数量，x_2为鲨鱼的数量，t为时间。微分方程为

$$\begin{cases}\dfrac{dx_1}{dt} = x_1(a_1 - b_1x_2)\\[2mm]\dfrac{dx_2}{dt} = -x_2(a_2 - b_2x_1)\end{cases}\tag{5.20}$$

式中a_1, a_2, b_1, b_2都是正常数。第一式鱼饵x_1的增长速度大体上与x_1成正比，即按a_1x_1速率增加，而被鲨鱼吃掉的部分按$b_1x_1x_2$的速率减少；第二式中鲨鱼的增长速度由于生存竞争的自然死亡或互相咬食按a_2x_2的速率减少，但又根据鱼饵的量的变化按$b_2x_1x_2$的速率增加。对$a_1=3$，$b_1=2$，$a_2=2.5$，$b_2=1$，$x_1(0)=x_2(0)=1$求解。画出解曲线图和相轨线图，可以观察到鱼饵和鲨鱼数量的周期振荡现象。

§5.7　补充知识：稳定性

1. Runge-Kutta法

由于Euler法对$[t_k, t_{k+1}]$上$y(t)$的平均斜率$K^* = (y(t_{k+1}) - y(t_k))/h$仅用左端点$t_k$的斜率作近似，精度较低。Runge-Kutta法利用$[t_k, t_{k+1}]$上多点斜率的加权平均作K^*的

近似。从而大大提高了计算精度。四阶 **Runge-Kutta** 格式为

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = f(t_n, y_n) \\ K_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}K_1\right) \\ K_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}K_2\right) \\ K_4 = f(t_n + h, y_n + hK_3) \end{cases} \tag{5.21}$$

它具有四阶收敛精度。

2. 数值算法稳定性

上述提到收敛精度是单步意义上的，计算中还涉及到误差的传播问题。不适当的步长可能导致计算误差恶性发展而使计算失败，这种现象称为**数值不稳定**。考虑一阶方程

$$y' = \lambda y, \ y(0) = 1 \tag{5.22}$$

其中 $\lambda < 0$。从而它的解 $y(t) = \exp(\lambda t)$ 恒正且渐近稳定。现取步长 h 用 Euler 法(5.13)式求解

$$y_{k+1} = (1 + \lambda h)y_k$$

当 $h < -1/\lambda$ 时，数值解正确反映了解函数性质；当 $-1/\lambda < h < -2/\lambda$ 时，数值解仍稳定但发生振荡；当 $h > -2/\lambda$ 时，数值解不稳定且振荡趋于无穷。一般地，对一阶方程组

$$\mathbf{y}' = A\mathbf{y} + \mathbf{b} \tag{5.23}$$

Euler 法数值稳定性条件为 $|1+\lambda h| < 1$，其中 λ 为 A 的任意特征值。四阶 Runge-Kutta 法数值稳定性条件稍宽但基本处于同一数量级。

3. 常微分方程解的稳定性

考虑一阶常微分方程初值问题

$$\begin{cases} y' = f(t, y), \ t > t_0 \\ y(t_0) = y_0 \end{cases} \tag{5.24}$$

称(5.24)式的特解 $\varphi(t)$是**稳定**的，若对 $\forall \varepsilon > 0$ 存在 $\delta > 0$，使当 $|y_0 - \varphi(t_0)| < \delta$ 时，以 y_0 为初值的解满足 $|y(t) - \varphi(t)| < \varepsilon (t > t_0)$；若还有 $|y(t) - \varphi(t)| \to 0 (t \to \infty)$，称 $\varphi(t)$是**渐近稳定**的。稳定性表明初值 y_0 的小扰动不会对解产生明显影响。微分方程描述的运动的轨线密切依赖于初值，而初值的计算或测定实际上不可避免地出现误差和干扰，如果描述运动的微分方程的特解是不稳定的，初值的微小误差或干扰，将招致"差之毫厘，谬以千里"的严重后果，因此不稳定的特解不宜作为设计的依据；反之，稳定的特解才是我们最感兴趣的。

例 7 考虑下列方程

$$y' = y - y^2 \tag{5.25}$$

它有两个常数特解(称为**驻定解**或**奇点**) $y(t) \equiv 0$ 和 $y(t) \equiv 1$。现在我们用数值解模拟初值

小扰动的影响。

解：先写 M 函数 eg5 _ 7fun. m，再用 M 文件 eg5 _ 7. m 作出在驻定解附近的解。结果表明 $y(t) \equiv 0$ 不稳定，而 $y(t) \equiv 1$ 稳定。

```
        %M 函数 eg5 _ 7fun. m
function f = eg5 _ 7fun(t, y)
f = y - y.^2; f = f(:);
        %M 文件 eg5 _ 7. m
clear; close; y0 = 0.01; h = 0.2;
for i = 1:8
    [t, y] = ode45('eg5 _ 7fun', [0, 10], y0);
    plot(t, y); hold on; y0 = y0 + h;
end
```

事实上，用 Symbolic 命令求解

```
»dsolve('Dy = y - y ^ 2', 'y(0) = a')
ans=
-1/(-1-exp(-t)*(1-a)/a)
```

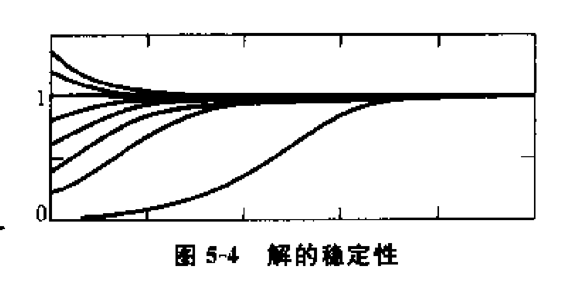


图 5-4 解的稳定性

可得 $y(t) = \dfrac{1}{1+\left(\dfrac{1}{y_0}-1\right)e^{-t}}$，所以 $y(t) \to 1(t \to \infty)$，如图 5-4 所示。这与上述图形描述一致。

对于(5.24)式的特解 $\varphi(t)$，令 $x(t) = y(t) - \varphi(t)$，那么 $x(t)$满足方程

$$x' = g(t, x),\ t > t_0 \tag{5.26}$$

其中 $g(t, x) = f(t, y) - f(t, \varphi)$，显然 $g(t, 0) = 0$，且与 φ 对应的特解为 $x \equiv 0$，从而稳定性问题总可归结为对于零解的稳定性问题。如对(5.25)式，特解 $y(t) \equiv 1$ 的稳定性与下述方程的零解稳定性等价。

$$x' = -x - x^2$$

对于一阶常系数线性微分方程组(5.23)。若 A 的特征值都具有负实部，(5.23)式的驻定解是稳定的；而当 A 有一个特征值具有正实部，(5.23)式的驻定解是不稳定的。非线性方程稳定性分析较困难，某些非线性方程可用局部线性化稳定性分析方法，较一般方法是 Lyapunov 直接法(参见文献[8])。

4. 动力系统的渐近性态

若微分方程组的右端不显含 t，即

$$y' = f(y) \tag{5.27}$$

则称为**自治方程组**或**驻定方程组**。自治方程组的解在相空间的几何表示为相轨线，也称**动力系统**。特别地，称相空间中使 $f(y) = 0$ 的点为**奇点**。线性动力系统(5.7)式的渐近性态较为简单，且一般与初值无关：(i)当 A 的特征值均具有负实部，趋于奇点；(ii)当 A 有正实部的特征值，趋于∞；否则(iii)当 A 的特征值均具有非正实部，趋于周期解。非线性动力系统的渐近性态，一般与初值有关，且会出现各种复杂的渐近性态。

(1)（奇点(0, 0)）$x' = -y - x^3$，$y' = x - y^3$

```
    function f = fun(t, y)
    f(1) = - y(2) - y(1) ^ 3; f(2) = y(1) - y(2) ^ 3; f = f(:);
》close; [t, y] = ode45('fun', [0, 1000], [0.9 0.6]');
》plot(y(:, 1), y(:, 2)); grid; hold on;
》plot(y(length(y), 1), y(length(y), 2), '.r')
```

(2)（无穷远）$x' = 2y^3$，$y' = 1/\sqrt{x}$

```
    function f = fun(t, y)
    f(1) = 2* y(2) ^ 3; f(2) = y(1) ^ ( - 0.5); f = f(:);
```

》命令同(1)。

(3)（周期解）$x' = y - 2(x^3/3 - x)$，$y' = -x$

```
    function f = fun(t, y)
    f(1) = y(2) - 2* (y(1) ^ 3/3 - y(1)); f(2) = - y(1); f = f(:);
```

》命令同(1)。

(4)（周期解）$x' = -0.5x - y$，$y' = -0.5y - z$，$z' = -0.5z - x$

```
    function f = fun(t, y)
    f(1) = - 0.5* y(1) - y(2); f(2) = - 0.5* y(2) - y(3);
    f(3) = - 0.5* y(3) - y(1); f = f(:);
》close; [t, y] = ode45('fun', [0, 30], [0.9 0.6 0.1]');
》plot3(y(:, 1), y(:, 2), y(:, 3)); grid; hold on;
》plot3(y(length(y), 1), y(length(y), 2), y(length(y), 3), '.r')
```

(5)（混沌）$x' = -3x + yz$，$y' = -10(y - z)$，$z' = -xy + 28y - z$

```
    function f = fun(t, y)
    f(1) = - 3* y(1) + y(2)* y(3); f(2) = - 10* (y(2) - y(3));
    f(3) = - y(2)* y(1) + 28* y(2) - y(3); f = f(:);
》close; [t, y] = ode45('fun', [0, 50], [0.9 0.6 0.1]');
》plot3(y(:, 1), y(:, 2), y(:, 3)); grid; hold on;
》plot3(y(length(y), 1), y(length(y), 2), y(length(y), 3), '.r')
```

5. 补充习题

10 编写四阶 Runge-Kutta 法程序并解习题 2。

11 对于例 4 分别取步长 $h = 0.001$，$h = 0.01$，$h = 0.05$ 用 Euler 法计算，分析误差产生的原因。

实验六　零件参数设计　随机模拟

本实验中我们学习随机模拟法(即 Monte Carlo 法)的基本原理、MATLAB 数据分析和随机模拟命令，并了解它们在随机变量模拟、积分计算和最优化计算等方面的应用。补充知识介绍了二项分布、Brown 运动模拟以及 Monte Carlo 法的盲点——小概率陷阱。

§6.1　引例:零件参数设计

一件产品由若干零件组装而成，标志产品性能的某个参数取决于这些零件的参数。零件参数包括标定值和容差两部分。进行成批生产时，标定值表示一批零件该参数的平均值，容差则给出了参数偏离其标定值的容许范围。若将零件参数视为随机变量，则标定值代表期望值，在生产部门无特殊要求时，容差通常规定为均方差的 3 倍。

粒子分离器某参数(记作 y)由 7 个零件的参数(记作 $x_1, x_2, \cdots, x_7$)决定，经验公式为

$$y = 174.42\left(\frac{x_1}{x_5}\right)\left(\frac{x_3}{x_2 - x_1}\right)^{0.85} \times \sqrt{\frac{1 - 2.62\left[1 - 0.36\left(\frac{x_4}{x_2}\right)^{-0.56}\right]^{\frac{3}{2}}\left(\frac{x_4}{x_2}\right)^{1.16}}{x_6 x_7}}$$

当各零件组装成产品时，如果产品参数偏离预先设定的目标值，就会造成质量损失，偏离越大，损失越大。y 的目标值(记作 y_0)为 1.50。当 y 偏离 $y_0 \pm 0.1$ 时，产品为次品，质量损失为 1000(元)；当 y 偏离 $y_0 \pm 0.3$ 时，产品为废品，损失为 9000(元)。问题是，要求对于给定的零件参数标定值和容差，计算产品的损失，从而在此基础上进行零件参数最优化设计。

在这个问题中，主要的困难是产品的参数值 y 是一个随机变量，而由于 y 与各零件参数间是一个复杂的函数关系，无法解析地得到 y 的概率分布。本实验采用随机模拟的方法计算。这一方法的思路其实很简单：用计算机模拟工厂生产大量“产品”(如 10000 件)，计算产品的总损失，从而得到每件产品的平均损失。这类试验如果用实物来做，由于需要大量人力物力而无法实现，但如果我们有了问题的数学模型，用计算机模拟方法就轻而易举。工程师们通常先用数值模拟得到一些结果，然后再根据经验和实物实验作决断。

§6.2　数学理论复习:概率论

自然界发生的现象可分为两类，一类现象在一定条件下发生的结果是完全可以预知的，称为**必然现象**。例如，从原点出发沿 x 轴正向以速度 v(m/s)作匀速直线运动的物体，必然在 t(s)后位于位置 vt(m)。另一类现象发生的结果在事先是无法准确预知的，称为**偶然现象**或**随机现象**。下面两个试验都是随机现象。

试验一：有 10 枚均匀硬币，随手抛在地上，有几枚正面向上?

试验二：按身份证号码随意挑 10 个中国男人，他们的平均身高是多少？

尽管随机现象的发生结果是不确定的，但还是有一定规律可循：试验一中正面向上的枚数一定是 0～10，5 枚向上的可能性比 8 枚向上可能性要大；试验二中平均身高基本在 1.40 米到 2.00 米之间，且在 1.70 米左右的可能性比 1.80 米左右的可能性要大。

一个随机事件 A 发生的可能性的大小，用一个介于 0 与 1 之间的数表示，称为 A 的概率，记为 $P(A)$。概率的意义在类似的现象大量重复发生时会表现出来。比如，在试验一中若 $P(5\text{ 枚向上}) = 0.25$，那么意味着"若把试验一做 100 遍，大致有 25 次左右出现 5 枚向上"。

随机现象中，变量的取值往往是不确定的，称为**随机变量**。描述随机变量取各种值的概率函数称为**概率分布**。对于随机变量，通常主要关心它的两个主要数字特征：**数学期望**(或称**均值**)用于描述随机变量的平均值，**方差**和**标准差**(或称**均方差**)用于描述随机变量分布的差异程度。另外，**协方差**和**相关系数**用于描述两个随机变量的线性关联程度。

随机变量的分布，根据其取值特点不同主要分为离散型和连续型两类。若用变量 ξ 表示试验一"正面向上次数"，其取值可能为 0, 1, 2, …, 10(离散点集)，则为离散型随机变量。典型的离散型分布有二项分布、Poisson 分布等。若用变量 η 表示试验二中"平均身高"，其取值可能为[1.0, 2.5]中任何值，则为连续型随机变量。典型的连续型分布有均匀分布、正态分布、指数分布、χ^2 分布、t 分布、F 分布等。

1. 二项分布

称一个随机试验为 Bernoulli 试验，若我们将试验可能结果描述成仅仅两个：A 发生或 A 不发生。设随机变量

$$\xi = \begin{cases} 1, & A\text{ 发生} \\ 0, & A\text{ 不发生} \end{cases}$$

那么 ξ 服从一个简单的离散型分布 $P(\xi = 1) = p$(A 发生的概率)，$P(\xi = 0) = 1 - p$。这称为 **Bernoulli 分布或 0-1 分布**。

将 Bernoulli 试验独立重复进行 n 次，称为 n 重 Bernoulli 试验。n 重 Bernoulli 试验中 A 发生的次数的分布为

$$P(\eta = k) = C_n^k p^k (1 - p)^{n-k},\ k = 0, 1, \cdots, n$$

称它为参数为 n、p 的**二项分布**，记为 $\xi \sim B(n, p)$。二项分布的数学期望 $E(\xi) = np$，方差 $D(\xi) = np(1 - p)$。

在试验一中，一枚硬币的试验结果只有两个：正面(A 发生)和反面(A 不发生)。由于 10 枚硬币的试验条件是类似的，且相互没有干扰，所以我们可以认为是一枚硬币的 10 次独立试验，那么正面向上的总枚数 $\xi \sim B(10, 0.5)$，其中 0.5 表示一枚均匀硬币正面向上的可能性为 50%。那么

$$P(\xi = 5) = C_{10}^5 0.5^5 0.5^5 = 0.2461,\ P(\xi = 8) = C_{10}^8 0.5^8 0.5^2 = 0.0439$$

可见 $P(\xi = 5)$ 远大于 $P(\xi = 8)$。

2. 均匀分布

连续型概率分布的表达方式与离散型有很大不同，因为连续型随机变量取值是无法列

举的，况且它在特定值上取值的概率总是 0。连续型概率分布是用**密度函数**来表示，它取值的概率可通过密度函数的积分得到。

均匀分布是一个简单而重要的连续型概率分布，其密度函数为

$$f(x)=\begin{cases}\dfrac{1}{b-a}, & x\in(a,b)\\ 0, & 其他\end{cases}$$

它的实际意义是，随机变量取值总是在$[a,b]$内，并且在每一点取值可能性相同。均匀分布的数学期望 $E(\xi)=(a+b)/2$，方差 $D(\xi)=(b-a)^2/12$。

3. 正态分布 $N(\mu,\sigma^2)$

正态分布是应用最广泛的概率分布，如图 6-1 所示。其密度函数为

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

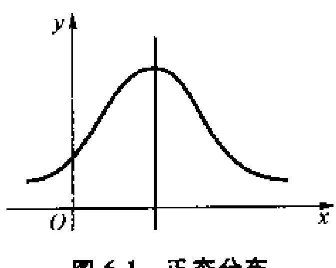


图 6-1 正态分布

记为 $N(\mu,\sigma^2)$，其中 μ 是随机变量取值的平均，而 σ 表征了随机变量取值的差异。特别地，$N(0,1)$称为**标准正态分布**。最典型的正态分布的例子是测量误差，另外像一大批学生的考试成绩、工业过程的噪声干扰等都近似服从正态分布。试验二中平均身高也近似服从正态分布。正态分布的数学期望 $E(\xi)=\mu$，方差 $D(\xi)=\sigma^2$。

正态分布有很多好的性质。设 $\xi_i\sim N(\mu_i,\sigma_i^2)$，$i=1,2,\cdots,n$，且这些随机变量相互独立，那么它的线性函数仍然是正态分布。特别地，若 $\xi\sim N(\mu,\sigma^2)$，那么$\dfrac{\xi-\mu}{\sigma}\sim N(0,1)$。

§6.3 随机模拟原理

关于随机分布的推断，一种理想化方法是将试验在相同条件下大量重复进行，对试验结果进行统计计算，从而得到概率分布和数字特征的结论，但是这样做需要耗费很大的人力和财力，况且很多随机现象实际上是无法再现的。随机模拟法也叫 **Monte Carlo 法**，它是用计算机模拟随机现象，通过大量仿真试验，进行分析推断，特别是对于一些复杂的随机变量，不能从数学上得到它的概率分布，而通过简单的随机模拟便可得到近似解答。Monte Carlo 法也用于求解一些非随机问题，如重积分、非线性方程组求解、最优化问题等。但需要指出的是，Monte Carlo 法计算量大，精度也不高，因而只适合一些用解析方法或常规数值方法难以解决问题的低精度求解，或用于对一些计算结果的验证。

大数定理 设随机变量 $\xi_1,\xi_2,\cdots,\xi_k,\cdots$相互独立，且 $E(\xi_i)=\mu$，$D(\xi_i)=\sigma^2$，那么这些随机变量的平均值依概率收敛于 μ，即对任意 $\varepsilon>0$，有

$$\lim_{k\to\infty}P\left(\left|\frac{1}{k}\sum_{i=1}^{k}\xi_i-\mu\right|<\varepsilon\right)=1$$

大数定理表明，当 $k\to\infty$ 时，样本平均值趋向于总体平均值，它是数理统计参数估计的理论基础，也是数字特征随机模拟的理论根据。

设 ξ 是一个分布已知的随机变量，为了求取 $\eta = f(\xi)$ 的概率分布或数字特征，生成 N 个（N 足够大）服从 ξ 的分布的随机数 $x_1, x_2, \cdots, x_N$，令 $y_i = f(x_i)$, $i = 1, 2, \cdots, N$，那么

$$P(\eta \in A) \approx \frac{A \text{ 中 } y_i \text{ 的个数}}{N}$$

$$E(\eta) \approx \overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

$$D(\eta) \approx s_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \overline{y})^2$$

其中 $\overline{y}$，s_y 分别称为 $y_i(i = 1, 2, \cdots, N)$ 的**样本均值**和**样本标准差**。

§6.4 数据分析和随机数生成的 MATLAB 命令

max, min	最大值，最小值；	bar	直方图；
sort	排序（从小到大）；	hist	数据分组及直方图；
mean	均值；	rand	[0, 1]均匀分布随机数；
median	中值；	randn	标准正态分布随机数；
std	标准差；	randperm	随机排列；
cov	协方差矩阵；	unidrnd	离散均匀分布随机数；
corrcoef	相关系数矩阵；	unifrnd	均匀分布随机数；
sum	各元素和；	normrnd	正态分布随机数；
cumsum	元素累计和；	binornd	二项分布随机数；
prod	各元素积；	poissrnd	Poisson 分布随机数；
cumprod	元素累计积；	mvnrnd	多维正态分布随机数。

其中自 unidrnd 以下的函数必须有统计工具箱(stats)支持。

1. 数据分析

数据分析函数 min, max, sort, mean, median, std, sum, prod, cumsum, cumprod 等标准用法都是对列状数据进行。

> a = min(X)　返回向量 X 的最小元素。
> X = min(A)　返回矩阵 A 每列最小元素构成的行向量。
> 其他函数用法类似。

例如

```
»data = [11    57    291
         13    54    278
         10    66    253
          9    46    307
         16    75    244
```

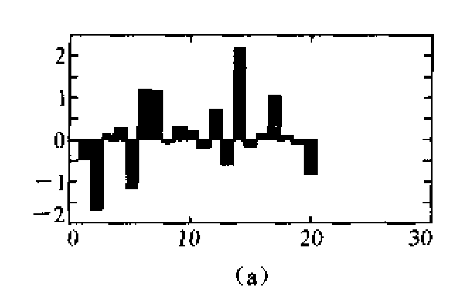
```
        15     70     256
         8     40     310];
》max(data)
ans=
   16    75    310
》mean(data), median(data)
ans=
   11.7143   58.2857   277.0000
ans=
   11    57    278
》std(data)
ans=
   3.0394   12.7895   26.7457
》sum(data)
ans=
          82        408        1939
》cumsum(data)
ans=
          11         57         291
          24        111         569
          34        177         822
          43        223        1129
          59        298        1373
          74        368        1629
          82        408        1939
》corrcoef(data) %将三列看成三个随机变量
ans=
      1.0000          0.8299        -0.7832
      0.8299          1.0000        -0.9633
     -0.7832         -0.9633         1.0000
```

2. 直方图

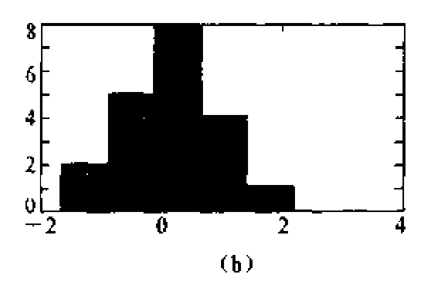
bar(Y) 作向量 Y 的直方图。
bar(X, Y) 作向量 Y 相对于 X 的直方图。
hist(X, k) 将向量 X 中数据等距分为 k 组，并作出直方图，缺省值为 k=10。
[N, X] = hist(Y, k) 不作图，N 返回各组数据个数，X 返回各组的中心位置。

```
》vdata = randn(1, 20); % 由于随机数的原因，以下每次结果略有不同
》subplot(1, 2, 1); bar(vdata); % (如图 6-2(a)所示)
```

```
»subplot(1, 2, 2); hist(vdata, 5); %(如图 6-2(b)所示)
»[n, x] = hist(vdata, 5)
n=
      2     5     8     4     1
x=
   -1.2807   -0.5110   0.2588   1.0286   1.7983
```



(a)



(b)

图 6-2 直方图

3. 随机数生成

R = rand(m, n) 生成(0, 1)上均匀分布的 m 行 n 列随机矩阵;
R = randn(m, n) 生成标准正态分布的 m 行 n 列随机矩阵;
P = randperm(N) 生成 1, 2, …, N 的一个随机排列;
R = unidrnd(N, m, n) 生成 1, 2, …, N 的等概率 m 行 n 列随机矩阵;
R = unifrnd(a, b, m, n) 生成[a, b]区间上的均匀分布 m 行 n 列随机数矩阵;
R = normrnd(mu, sigma, m, n) 生成均值为 mu,均方差为 sigma 的 m 行 n 列正态分布随机数矩阵;
R = binornd(k, p, m, n) 生成参数为 k, p 的 m 行 n 列正态分布随机数矩阵。它模拟在 k 次重复试验中某事件(发生概率为 p)出现的次数;
R = mvnrnd(mu, sigma, m) 生成 n 维正态分布数据,这里 mu 为 n 维均值向量,sigma 为 n 阶协方差矩阵(它必须是正定的),R 为 $m\times n$ 矩阵,每行代表一个随机数。

例如(由于随机数的原因,以下每次结果略有不同),

```
»a = rand(1, 1000); [mean(a), std(a)]
ans =
    0.5013     0.2882          % 均值 0.5,标准差 1/2/sqrt(3)
»b = randn(1, 1000); [mean(b), std(b)]
ans =
    0.0597     1.0396          % 均值 0,标准差 1
»c = normrnd( - 5, 6, 1, 1000); [mean(c), std(c)]
ans=
   -5.0679     6.0592          % 均值 -5,标准差 6
»r = mvnrnd([0, 0]', [1, 0.9; 0.9, 1], 100);
```

```
》plot(r(:, 1), r(:, 2), 'o'); % 如图 6-3
```

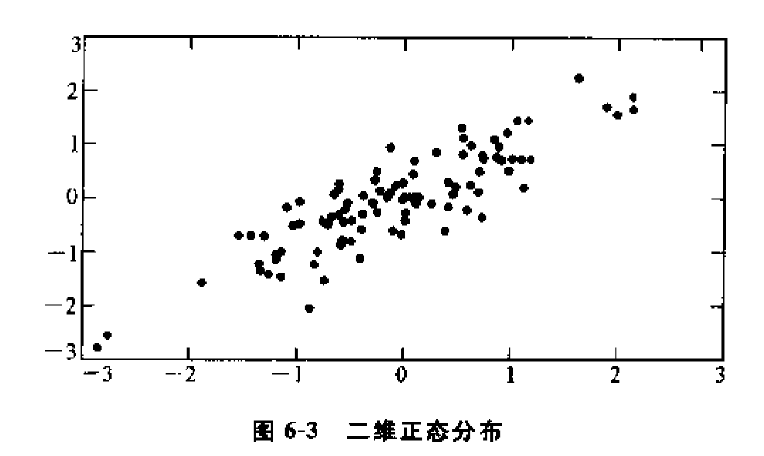


图 6-3　二维正态分布

注：MATLAB 有两个基本的随机数产生函数 rand 和 randn。它们允许用户自己设置随机数种子，若将种子设置为系统时间，

```
》rand('seed', sum(100*clock))
```

则可得到真正意义上的随机试验。由于内存或速度的限制，矩阵阶数不宜太大（一般在 10^5 以下）。

§6.5　实验例题

例 1　（零件参数设计）表 6-1 给定引例中某设计方案 7 个零件参数标定值及容差。容差分为 A、B、C 三个等级，用与标定值的相对值表示，A 等为 ±1%，B 等为 ±5%，C 等为 ±15%。

表 6-1　零件参数的标定值和容差

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
标定值	0.1	0.3	0.1	0.1	1.5	16	0.75
容　差	B	B	B	C	C	B	B

求每件产品的平均损失。

解：可以假设 7 个零件参数服从正态分布。根据上表及标定值和容差的定义，$x_1 \sim N(0.1, (0.005/3)^2)$，$x_2 \sim N(0.3, 0.005^2)$，$x_3 \sim N(0.1, (0.005/3)^2)$，$x_4 \sim N(0.1, 0.005^2)$，$x_5 \sim N(1.5, (0.225/3)^2)$，$x_6 \sim N(16, (0.8/3)^2)$，$x_7 \sim N(0.75, (0.0375/3)^2)$，下面的 M 脚本 eg6_1.m 产生 1000 对零件参数随机数，通过随机模拟法求得近似解约 $f=2900$ 元。

```
        %M 脚本 eg6_1.m
clear; mu=[.1, .3, .1, .1, 1.5, 16, .75];
sigma=[.005/3, .005, .005/3, .005, .225/3, .8/3, .0375/3];
for i=1:7
    x(:, i)=normrnd(mu(i), sigma(i), 1000, 1);
end
```

```
t = (1 - 2.62* (1 - 0.36* (x(:, 4)./x(:, 2)).^( - 0.56)).^1.5...
.* (x(:, 4)./x(:, 2)).^1.16)./x(:, 6)./x(:, 7);
y = (x(:, 1)./x(:, 5)).* (x(:, 3)./(x(:, 2) - x(:, 1))).^0.85;
y = 174.42* y.* t.^0.5;
d = abs(y - 1.5);
f = sum(9000* (d>0.3) + 1000* (d<=0.3).* (d>0.1))/1000
```

例 2 (Monte Carlo 法计算积分)随机模拟法也可用于求解一些复杂的非随机问题。考虑二重积分

$$I = \iint_A f(x, y)\mathrm{d}x\mathrm{d}y,\ 其中\ f(x, y) \geqslant 0,\ \forall (x, y) \in A$$

根据其几何意义,它是以 $f(x, y)$为曲面顶,A 为底的柱体 C 的体积。用下列简单思路求 I 的近似值:假设 C 被包在几何体 D 的内部,D 的体积为已知,若在 D 内产生 1 个均匀分布的随机数,那么

$$P(随机数落在\ C\ 内) \approx C\ 的体积/D\ 的体积$$

现产生在 D 内 N 个均匀分布的随机数,若其中 N_C 个在 C 的内部,那么

$$I \approx D\ 的体积 \times N_C/N$$

下面我们用这一方法计算实验四例 1 积分

$$I = \iint_{x^2+y^2\leqslant 1} \sqrt{1-x^2}\,\mathrm{d}x\mathrm{d}y$$

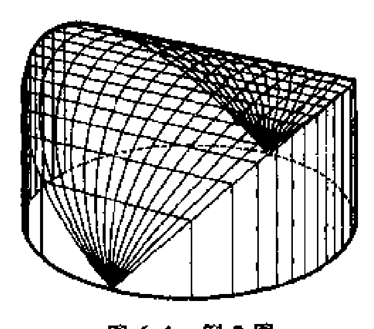


图 6-4 例 2 图

显然该几何体在立方体 $|x|\leqslant 1,\ |y|\leqslant 1,\ 0<z<1$ 的内部,立方体体积为 4。面 C 是 $x^2+y^2\leqslant 1,\ x^2+z^2\leqslant 1,\ z>0$。M 文件 eg6_2.m 中 N 取 1000,计算得 $I\approx 2.624$。由于随机模拟的特点,每次运行的结果都有差异。N 越大结果越精确。如图 6-4 所示。

```
%M 脚本 eg6_2.m
clear; N = 1000;
x = unifrnd( - 1, 1, N, 1);
y = unifrnd( - 1, 1, N, 1); z = rand(N, 1);
c1 = (x.^2 + y.^2)<1; c2 = (x.^2 + z.^2)<1;
Nc = sum(c1&c2); I = Nc/N* 4
```

例 3 (最优化计算 Monte Carlo 法)用随机模拟法。求下列函数的最大值

$$f(x) = (1-x^3)\sin(3x),\ -2\pi < x < 2\pi$$

解:为了便于理解,我们先作图 6-5,

```
»clear; close; x = - 2* pi:0.001:2* pi;
»f = (1 - x.^3).* sin(3* x); plot(x, f)
```

可见函数在 −6 和 6 附近达到最大值约 200。但是,使用优化算法

```
》fun = '-(1-x.^3).*sin(3*x)';
》x = fmin(fun, -2*pi, 2*pi), f = -eval(fun)
x=
  -3.7505
f=
  52.0046
```

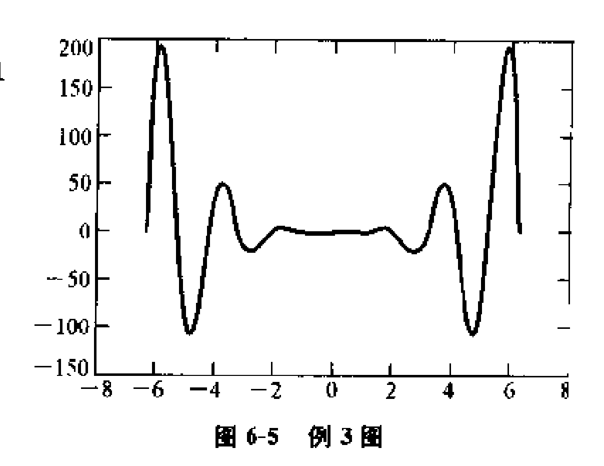
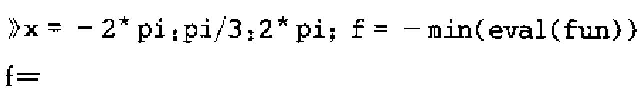


图 6-5 例3图

显然结果是错误的，原因是 fmin 容易陷入局部极值。这也是许多其他优化算法难以克服的一个困难。若用规则格点搜索法而步长选取不当，例如取 h = pi/3,



```
》x = -2*pi:pi/3:2*pi; f = -min(eval(fun))
f=
  1.8299e-013
```

由于函数的周期性造成一个荒唐的结果。现在我们用随机模拟，就是随机产生若干个自变量的值来搜索，我们取 13 个点(与规则格点搜索法所用采样点数一致)

```
》x = unifrnd(-2*pi, 2*pi, 13, 1); f = -min(eval(fun))
f=
  185.0419
```

结果要好得多。Monte Carlo 法的主要优点在于它是一个全局优化算法。

例 4 (中心极限定理)设 $\xi_1, \xi_2, \cdots, \xi_k, \cdots$ 独立同分布且 $E(\xi_i)=\mu$, $D(\xi_i)=\sigma^2$，则当 k 很大时，$\eta_k=\sum_{i=1}^{k}\xi_i$ 近似服从 $N(k\xi, k\sigma^2)$。

中心极限定理表明，大量独立随机变量的和近似服从正态分布，它是正态分布应用的理论依据。下列 M 文件 eg6_4.m 给出 100 个(0, 1)上独立均匀分布的和的分布。

%M 脚本 eg6_4.m

```
clear; close; K = 100;
N = K; M = 100; r = rand(N, M);
s = sum(r);
mu = mean(s)
sigma = std(s)
[n, x] = hist(s, mu-5*sigma: sigma: mu+5*sigma);
bar(x, n/M/sigma, 'r'); hold on;
h = mu-5*sigma: 0.1*sigma: mu+5*sigma;
t = exp(-(h-mu).^2/2/sigma^2)/sqrt(2*pi)/sigma;
plot(h, t, 'k'); title('中心极限定理');
legend('独立 RV 和','正态分布'); hold off;
```

§6.6 实验习题

1 (掷硬币)考虑将一枚均匀硬币掷 N 次,当 N 很大时,正面出现的概率接近 0.5,设计一个随机模拟试验显示这一现象。

2 (均匀分布)利用 rand 编写一个 M 函数 uni_rnd.m,使之能生成任意区间$[a, b]$上的均匀分布随机数 $m\times n$ 矩阵。使用格式为 uni_rnd(a, b, m, n)。

3 (正态分布)利用 randn 编写一个 M 函数 norm_rnd.m,使之实生成的正态分布 N(mu, sigma²)随机数 $m\times n$ 矩阵。使用格式为 norm_rnd(mu, sigma, m, n)。

4 (二项分布的正态近似)Demorvie-Laplace 中心极限定理指出,若 $\eta\sim B(n, p)$, n 很大,则规范化随机变量

$$\frac{\eta-np}{\sqrt{np(1-p)}}\text{ 近似服从 } N(0, 1)$$

用实验进行验证。

5 用 Monte Carlo 法计算积分

$$\int_0^1\frac{\exp\left(-\frac{x^2}{2}\right)}{\sqrt{2\pi}}dx,\ \int_0^{2\pi}\exp(x/2)\sin^2(x)dx,\ \int_0^{\pi}\int_0^{\sin(x)}\exp(-x^2-y^2)dxdy$$

6 分别用 Monte Carlo 法和 fmins 求下列二元函数最大值,并通过图形作出评论。

$$f(x, y)=(x^2+2y^2+xy)\exp(-x^2-y^2),\ |x|<1.5,\ |y|<1.5$$

7 设计一个随机模拟实验来验证大数定理。

§6.7 补充知识:小概率陷阱

1. 二项分布随机数产生

如何用最基本的随机数函数 rand 产生二项分布 $B(n, p)$的一个随机数呢?先考虑 Bernoulli 试验,为此产生一个(0, 1)上均匀分布随机数,若这个数小于 p,则试验结果记为 1,否则记为 0,那么试验结果服从 0-1 分布。n 个 0-1 分布随机数的和便是一个二项分布随机数。下列 M 函数 bion_rnd.m 产生二项分布随机数,使用格式

r = bino_rnd(n, p, mm, nn)

r 返回一个由二项分布 $B(n, p)$随机数构成的 mm×nn 阶矩阵。

```
          %M 函数 bion_rnd.m
function r = bino_rnd(n, p, mm, nn)
r = zeros(mm, nn);
if(p<0|p>1|n<0|n~ = round(n))
    warning('invalid parameter');
end
```

```
for i = 1: n, u = rand(mm, nn); r = r + (u<p); end
```

事实上，统计工具箱函数 binornd 与上述函数构造法基本一致。

2. 小概率陷阱

随机试验法对零概率现象或小概率事件(往往出现于退化情形)往往会做出错误结论，考虑一个简单的约束优化问题

例 5　max　$f(x) = x_1x_2x_3$

$$\begin{cases} -x_1 + 2x_2 + 2x_3 \geqslant 0 \\ x_1 + 2x_2 + 2x_3 \leqslant 72 \\ 10 \leqslant x_2 \leqslant 20 \\ x_1 - x_2 = 10 \end{cases}$$

显然有 $20 \leqslant x_1 \leqslant 30$，$10 \leqslant x_2 \leqslant 20$，$-10 \leqslant x_3 \leqslant 16$。编写 M 文件 eg6_5a.m

```
            %M 文件 eg6_5a.m
clear; N = 50;
x10 = []; x20 = []; x30 = []; vmax = -inf;
x1 = unifrnd(20, 30, N, 1);
x2 = unifrnd(10, 20, N, 1);
x3 = unifrnd(-10, 16, N, 1);
for i1 = 1: N
    for i2 = 1: N
        for i3 = 1: N
            if -x1(i1) + 2*x2(i2) + 2*x3(i3)>= 0&...
                x1(i1) + 2*x2(i2) + 2*x3(i3)<= 72&...
                x1(i1) - x2(i2) == 10,
            v = x1(i1)*x2(i2)*x3(i3);
            if v>vmax,
                vmax = v; x10 = x1(i1); x20 = x2(i2); x30 = x3(i3);
end; end; end; end; end
x = [x10, x20, x30], vmax
```

经过长时间计算会得出一荒唐结论 vmax=−inf。什么原因呢？由于约束 $x_1 - x_2 = 10$ 的影响使得可行域(即约束条件界定的区域)在三维立方体内是亏维的，体积为零。这样三维随机数进入可行域概率为零(约束 $x_1 - x_2 = 10$ 无法满足)，从而找不到可行解，最优解也就无从谈起了。

若将"$x_1 - x_2 = 10$"改为"$|x_1 - x_2 - 10| < 1e-3$"从而转化为正概率问题，则可求解。这里 1e−3 代表一个很小的误差限。解得 vmax = 2662。但这样的求解效率太低。

现在我们将问题转化为下列二维的，

$$\max f = (x_2 + 10)x_2x_3$$

$$s.t.\ x_2 + 2x_3 \geqslant 10,\ 3x_2 + 2x_3 \leqslant 62,$$

$$10 \leqslant x_2 \leqslant 20,\ -5 \leqslant x_3 \leqslant 16$$

编写 M 文件 eg6 _ 5b. m 求解得 x = (21.6584, 11.6584, 13.3831), vmax = 3379。而计算量只有上面的 50 分之一。

```
            %M 文件 eg6 _ 5b. m
clear; N = 50;
x20 = []; x30 = []; vmax = - inf;
x2 = unifrnd(10, 20, N, 1);
x3 = unifrnd(-5, 16, N, 1);
for i2 = 1:N
    for i3 = 1:N
        if x2(i2) + 2*x3(i3)) = 10&...
              3*x2(i2) + 2*x3(i3)( = 62
         v = (x2(i2) + 10)*x2(i2)*x3(i3);
         if v)vmax,
              vmax = v; x20 = x2(i2); x30 = x3(i3);
         end; end; end; end;
x = [x20 + 10, x20, x30], vmax
```

Monte Carlo 法计算量大,精度也不高,因而只适合一些用解析方法或常规数值方法难以解决问题的低精度求解,或用于对一些计算结果的验证。在使用 Monte Carlo 法时,要注意克服小概率陷阱。

3. Brown 运动

Brown 运动是英国植物学家 Brown 在观察液体中浮游微粒运动时发现的随机现象,现在已成为随机过程理论最重要的概念之一。

(1) 一维 Brown 运动。下列 M 函数 brwnm. m 给出了一维 Brown 运动,使用格式

$$[t, w] = brwnm(t0, tf, h)$$

其中[t0, tf]为时间区间,h 为采样步长,w(t)为 Brown 运动(或称 Wiener 过程)。

```
        %M 函数 brwnm. m
    function [t, w] = brwnm(t0, tf, h)
    t = t0: h: tf; t = t'; x = randn(size(t));
    w(1) = 0; for k = 1: length(t) - 1, w(k + 1) = w(k) + x(k); end;
    w = sqrt(h)*w; w = w(:);
```

(2) 若 $w_1(t)$, $w_2(t)$都是一维 Brown 运动且相互独立,那么($w_1(t)$, $w_2(t)$)是一个二维 Brown 运动。给出二维 Brown 运动模拟。下列 M 文件 brwn2. m 给出 20 颗微粒的扩散过程显示。

```
                    %M 脚本 brwn2. m
clear; close;
t0 = 0; tf = 1; h = 0.01; x = zeros((tf - t0)/h + 1, 20);
for j = 1:20,
    [t, x(:, j)] = brwnm(t0, tf, h);
```

```
        [t, y(:, j)] = brwnm(t0, tf, h);
    end
    for i = 1:length(t),
        plot(x(i, 1:19), y(i, 1:19), '.', 'markersize', 24);
        hold on;
        plot(x(i, 20), y(i, 20), 'r.', 'markersize', 24);
        axis([-3 3 -3 3]); grid on; hold off;
        pause(1);
    end
```

4. 补充习题

8 （χ^2分布）设 ξ_1，ξ_2，…，ξ_v 独立且服从 $N(0, 1)$，则 $\eta = \sum_{i=1}^{v}\xi_i^2$ 服从自由度为 n 的 χ^2 分布，记为 $\eta \sim \chi^2(v)$。编写一个 M 函数 chi2_rnd(m, n, v)，使能产生自由度为 v 的 χ^2分布 $m \times n$ 随机矩阵，然后作出自由度 4 的 χ^2 分布密度的直方图。

9 考虑下列命题

“任何二阶方阵都是可逆的”

很明显，这是一个错误命题。例如

$$\begin{pmatrix} 1 & 2 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ -2 & -4 \end{pmatrix}$$

都是不可逆的。现在若使用 Monte Carlo 法，设计如下试验：

在 realmin 和 realmax 之间随机任取一个 2×2 矩阵，检查其行列式，若行列式等于 0，则找到反例，停止；否则重新取一个；若取了 10000 个矩阵仍然找不到，则认为无不可逆阵。编写程序实现上述试验，看出什么问题？考虑怎样改造实验，才可找到不可逆方阵？

10 作出三维 Brown 运动模拟图。

实验七　身高、体重与体育成绩　统计推断

本实验中我们将学习用 MATLAB 统计工具箱(stats)命令求解参数估计、假设检验和多元线性回归等统计推断问题,并通过一些实例了解数理统计的应用。补充知识介绍了数据预处理、小样本检验、分布拟合度检验、独立性检验和非线性回归等。

§7.1　引例:学生的身高、体重与体育成绩

从某寄宿制中学高三学生中随机抽取 100 名男生的身高、体重(单位分别为 cm 和 kg)和体育课成绩如表 7-1 所示。现要求

表 7-1　学生的身高、体重与体育成绩数据

身 高	体 重	成 绩	身 高	体 重	成 绩	身 高	体 重	成 绩	身 高	体 重	成 绩
167	50	85	179	69	86	174	61	67	174	61	65
179	63	93	172	59	84	166	48	76	173	65	76
168	54	78	159	59	81	167	55	67	159	48	72
187	79	91	167	56	83	177	65	67	170	68	82
173	62	68	163	51	66	162	62	73	179	66	96
176	70	86	158	44	70	182	70	84	167	51	87
170	57	81	175	69	69	171	60	87	174	65	70
170	57	76	174	61	86	175	58	75	170	52	62
162	53	71	159	53	79	179	66	71	175	66	84
177	67	67	160	47	88	172	61	85	168	55	76
179	68	75	168	63	88	174	61	85	163	57	88
172	61	83	169	53	81	160	45	70	170	63	71
170	58	84	171	63	85	164	57	70	177	62	79
177	67	79	160	51	81	176	60	75	172	58	85
172	62	87	165	53	67	182	73	80	167	58	87
166	53	81	174	63	87	173	59	73	169	59	88
174	62	83	164	53	64	169	51	92	171	63	80
171	63	63	180	73	83	171	54	84	179	65	79
169	56	76	170	58	83	175	63	70	168	58	70
167	64	85	175	68	86	173	65	81	160	45	83
169	64	71	158	55	80	175	72	68	165	52	63
166	53	79	174	67	82	162	47	63	167	53	75
163	50	80	162	50	85	172	62	89	164	56	89
175	66	74	172	63	65	170	63	77	156	45	72
173	66	91	166	50	86	171	68	80	166	50	69

(1) 给出这些数据直观的图形描述;

(2) 根据这些数据对全校学生的平均身高和体重作出估计;

(3) 若普通中学同龄男生平均身高为168.3cm，平均体重56.2kg，能否认为该中学男生身高和体重与普通中学相比有显著区别？

(4) 身高和体重对体育成绩有何影响？

§7.2 数学理论复习：数数统计的基本概念

科学研究的目的是发现事物的规律性，经典的问题可通过物理定律建立机理上的规律，而缺乏本质规律认识的问题，往往要通过对事物发生现象的分析来进行推断。数据是事物现象的反映，是科学推断的具体依据。例如在国民经济的投入产出分析中，消耗矩阵往往是未知的，那么我们可以利用投入量和产出量的几组记录通过求解矩阵方程求得消耗矩阵，即国民经济各部门间的依赖规律。但是由于现实世界的复杂性，这样的分析往往是不可靠的，因为数据测量带有误差，而模型本身也常常是理想化的。如用线性近似非线性，定常近似时变，次要因素的忽略，不可测因素的忽略等都会影响到结论的有效性。从数理统计学观点来看，测量数据是概率分布的一个抽样，具有一定的随机性(有些模型本身就是随机分布)，我们的任务是从数据中分高出随机因素的成分，从而挖据出事物规律性的成分。进行这样的分析建立在收集大量数据的基础上，称为**统计分析**。

1. 总体和样本

所谓**总体**(或称**母体**)就是一大批具有特定意义的待分析的随机数据，数学上用一个未知的概率分布表示，但在多数论况下，它的分布类型是已知的，只是某些参数未知。例如总体服从正态分布 $N(\mu, \sigma^2)$，其中 μ, σ^2 未知。总体的一部分数据 $x_1, x_2, \cdots, x_n$ 称为一个容量为 n 的**样本**或**子样**。统计推断就是根据样本(总体的一部分)对总体进行推断。但怎样获取样本会直接影响统计推断的结果，理想的样本是相互独立的且与总体同分布。

2. 统计量

不含未知参数的样本的函数称为统计量，它是样本特征的反映，选取一个正确的统计量是统计推断的关键，几个最基本的统计量是

(1) **样本均值** $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ 反映了样本取值的中心。

(2) **样本中位数** 将 $x_1, x_2, \cdots, x_n$ 从小到大顺序排为 $x_{(1)}, x_{(2)}, \cdots, x_{(n)}$，那么若 $n = 2k-1$，中位数为 $x_{(k)}$，若 $n = 2k$，中位数为 $(x_{(k)} + x_{(k+1)})/2$。中位数也是样本中心特征，它是这样一个值：比它大的值与比它小的值一样多。

(3) **样本方差** $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$，**样本标准差** $s = \sqrt{s^2}$ 反映了样本的对于均值的偏离程度。**样本极差** $x_{(n)} - x_{(1)}$ 也是离散程度的反映。

(4) **样本协方差** $\mathrm{cov}(x, y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$，**样本相关系数** $r(x, y) = \frac{\mathrm{cov}(x, y)}{s_x s_y}$。相关系数反映了样本 $x_1, x_2, \cdots, x_n$ 与样本 $y_1, y_2, \cdots, y_n$ 的线性相关关系，其中 s_x, s_y 为样本标准差。若 r 接近1，x 较大时 y 也较大；若 r 接近-1说明 x 较大时 y 较小；若 r 接近0说明 x 与 y 取值大小无线性相关关系，总之 $|r|$ 接近1说明线性关系密切。

当 n 足够大时，样本均值和样本标准差结合起来大致地推述了数据的分布结构：大约

68%落在 $(\bar{x}-s, \bar{x}+s)$ 之内，而有大约 95%在 $(\bar{x}-2s, \bar{x}+2s)$ 之内，几乎所有数据在 $(\bar{x}-4s, \bar{x}+4s)$ 之间。

3. 抽样分布及其 p 分位数

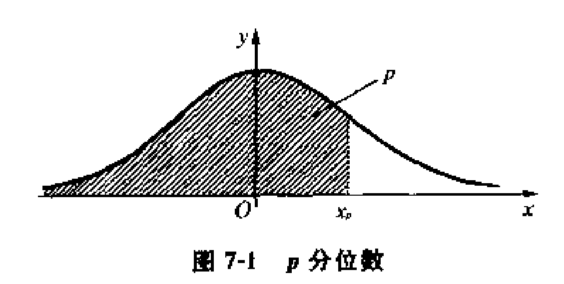


图 7-1　p 分位数

统计量的概率分布称为抽样分布，最常用的抽样分布有 z 分布(即标准正态分布)、t 分布、χ^2 分布、F 分布等。一个随机变量 ξ 的 p 分位数 x_p 定义为

$$P(\xi < x_p) = p, 0 \leqslant p \leqslant 1$$

如图 7-1 所示。

§7.3　MATLAB 统计分析工具统

MATLAB 的统计分析工具箱 stats 提供了丰富的统计分析函数，除了实验六介绍的随机数产生外，还有概率分布、参数估计、假设检验、线性和非线性模型、试验设计等。下面仅介绍一些本实验用到的内容，其他参见附录。

normpdf	正态分布密度；	ttest	t 检验；
normcdf	正态分布分布函数；	ttest2	双样本 t 检验；
norminv	正态分布 p 分位数；	regress	回归分析；
tinv	t 分布 p 分位数；	rcoplot	回归分析残差图；
chi2inv	χ^2分布 p 分位数；	nlinfit	非线性回归；
finv	F 分布 p 分位数；	nlparci	非线性回归箱度；
normfit	正态分布参数估计；	nlpredci	非线性回归预测。
ztest	z(正态)检验；		

1. 正态分布

y = normpdf(x, mu, sigma)　返回参数为 mu 和 sigma 的正态分布密度函数在 x 处的值，即

$$y = \frac{1}{\sqrt{2\pi}\text{sigma}}\exp\left(-\frac{(x-\text{mu})^2}{2\text{sigma}^2}\right)。$$

p = normcdf(x, mu, sigma)　正态分布函数值，即 $p = \int_{-\infty}^{x} \text{normpdf}(t, \text{mu}, \text{sigma})dt$。

x = norminv(p, mu, sigma)　normcdf 的逆函数，即 p 分位数。

例如(如图 7-2 所示)，

```
»a = normpdf(90, 80, 10) - normpdf(70, 80, 10)
   a=
        0
»h = normcdf(90, 80, 10) - normcdf(70, 80, 10)
   b=
```

```
    0.6827
》p=(1-b)/2; c=norminv(p, 80, 10)
  c=
    70
》d=norminv(1-p, 80, 10)
  d=
    90
```

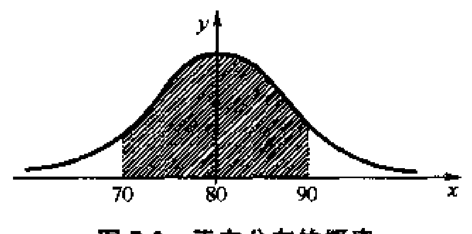


图 7-2　正态分布的概率

2. t 分布，χ^2 分布，F 分布

x=tinv(p, n)　自由度为 n 的 t 分布 p 分位数。 x=chi2inv(p, n)　自由度为 n 的 χ^2 分布 p 分位数。 x=finv(p, m, n)　自由度为 m，n 的 F 分布 p 分位数。

例如

```
》x1=tinv(0.05, 10), x2=tinv(0.95, 10)
x1=
  -1.8125
x2=
  1.8125
```

§7.4　统计推断方法

1. 参数估计

在统计推断中，总体参数 θ 未知，需要根据样本 $x_1, x_2, \cdots, x_n$ 估计 θ 的值。参数估计分为两类：点估计和区间估计。**点估计**就是直接给出 θ 的估计值，如"θ 大约等于 1.3"。但点估计缺乏对估计精度的说明。**区间估计**给出 θ 的估计值区间，并附加一个概率，如"θ 的 95%置信区间是[1.24, 1.36]"，含意是：θ 在[1.24, 1.36]内的概率为 0.95。

设有总体 $F(x, \theta)$，其中参数 θ 未知，现有来自 $F(x, \theta)$ 的一个样本 $x_1, x_2, \cdots, x_n$，要估计 θ 的值。如有区间 $CI=[\theta_1, \theta_2]$，使得

$$P(\theta_1 < \theta < \theta_2) = 1-\alpha$$

称 CI 为 θ 的 $100(1-\alpha)\%$**置信区间**。

设 ξ 为正态总体 $N(\mu, \sigma^2)$，μ, σ 未知，$x_1, x_2, \cdots, x_n$ 为样本，那么 μ 和 σ 的点估计分别为

$$\mu = \bar{x},\ \sigma = s$$

它们的 $100(1-\alpha)\%$ 的置信区间分别为

$$\left(\bar{x} - t_{1-\frac{\alpha}{2}}(n-1)\frac{s}{\sqrt{n}},\ \bar{x} + t_{1-\frac{\alpha}{2}}(n-1)\frac{s}{\sqrt{n}}\right)$$

$$\left(\frac{(n-1)s^2}{\chi^2_{1-\frac{\alpha}{2}}(n-1)}, \frac{(n-1)s^2}{\chi^2_{\frac{\alpha}{2}}(n-1)}\right)$$

stats 函数 normfit 完成上述估计，其使用格式

```
[muhat, sigmahat, muci, sigmaci] = normfit(x, alpha)
```

其中

x：列状样本数据；

alpha：α 值；

muhat：μ 的点估计；

sigmahat：σ 的点估计；

muci：μ 的置信区间；

sigmaci：σ 的置信区间。

例如，

```
» x = randn(20, 7); [a, b, aci, bci] = normfit(x, 0.1)
a=
   -0.2252    0.2460   -0.3470    0.2509   -0.0324   -0.4589    0.1966
h=
    0.8829    1.1438    1.0482    0.8670    1.0260    0.6847    0.9873
aci=
   -0.5666   -0.1963   -0.7523   -0.0844   -0.4291   -0.7236   -0.1851
    0.1162    0.6882    0.0583    0.5861    0.3643   -0.1942    0.5784
bci=
    0.7010    0.9081    0.8322    0.6883    0.8146    0.5436    0.7839
    1.2100    1.5675    1.4365    1.1882    1.4060    0.9383    1.3531
```

理论上 $a=0$，$b=1$，结果约有 90%置信区间套住真实值。本例给出 7 组估计值，其中 6 对套住真实值。

2. 假设检验

许多统计推断常涉及对某假设的正确性作出“是”与“否”的判决，例如某厂产品是否合格，某数学模型是否与现实相符等。在这类问题中，我们往往是要判断手头的数据是否与某假设(称为**零假设或原假设** H_0)明显不符，所以也称为**显著性检验**。根据不利于 H_0 的数据偏向提出与之对立的假设(称为**备择假设或对立假设** H_1)。在统计检验中，如果你想分析某事件是否明显，那么通常此事件是 H_1。结论常如“在显著性水平 $\alpha=0.05$ 下，拒绝 H_0”，含义是：推断为“数据与 H_0 明显不符”，这一统计推断结论可能错误，但错误概率只有 0.05。

设 $x_1, x_2, \cdots, x_n$ 为来自正态总体 ξ 的样本，其均值 μ 未知，方差 σ^2 已知，

$$H_0: \mu=\mu_0,\ H_1: \mu\neq\mu_0 (\text{或} \mu>\mu_0,\ \mu<\mu_0)$$

检验统计量 $Z=\dfrac{\overline{x}-\mu_0}{\sigma/\sqrt{n}}\sim N(0,1)$，拒绝域 $|Z|>u_{1-\alpha/2}$ (或 $Z>u_{1-\alpha}$，$Z<u_\alpha$)。如图 7-3

所示。即使总体非正态或 σ 未知，只要样本容量 n 相当大（一般要求大于 45），仍可以样本标准差 s 代替 σ 计算。

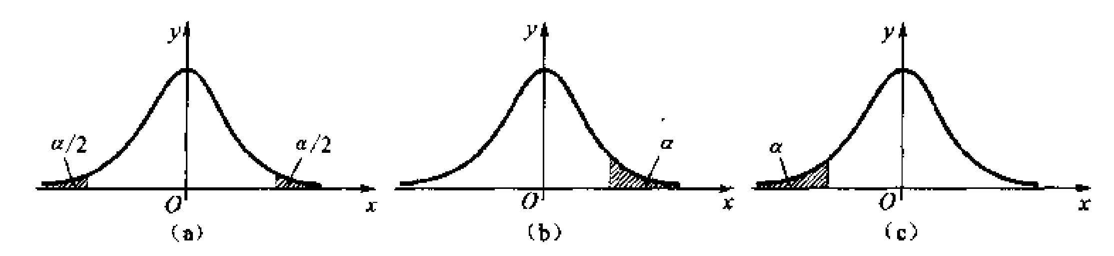


图 7-3 各种假设检验

(a) 双边 (b) 右边 (c) 左边

stats 函数 ztest 给出 z 检验法。使用格式

```
[h, sig] = ztest(x, m, sigma, alpha, tail)
```

其中

x：样本列向量；

m：μ_0；

sigma：σ；

tail：0 表示双边检验 ($H_1: \mu \neq \mu_0$)（缺省值）；

1 表示右边检验 ($H_1: \mu > \mu_0$)；

−1 表示左边检验 ($H_1: \mu < \mu_0$)；

alpha：显著性水平 α（缺省值 0.05）；

h：返回 1 表示拒绝 H_0，返回 0 表示接受 H_0；

sig：返回临界值拒绝概率，sig < alpha 时 $h = 1$。

例 1 （比例检验）某外商声称，他提供给工厂的某种零件至少有 95% 是符合规范的。现测试了 200 台这种设备，发现有 15 合是不符合规范的。在显著性水平 $\alpha = 0.05$ 下，能否相信该外商的声称是真实的？

解：设总体是一个 0-1 分布随机变量

$$\xi = \begin{cases} 0, & \text{不符合规范} \\ 1, & \text{符合规范} \end{cases}, \quad P(\xi = 1) = p$$

样本值可视为 18 个 0，182 个 1。

$$H_0: p = 0.95, \quad H_1: p < 0.95$$

由于样本容量 200 很大，可使用大样本 Z 检验 ztest。

```
» clear; x = [ones(1, 185), zeros(1, 15)];
» [h, sig] = ztest(x, 0.95, std(x), 0.05, -1)
h=
    0
sig=
```

0.0903

据此，接受 H_0，即从统计意义上采信该外商的声称。

3. 线性回归

设有多元线性回归模型

$$y = b_1 x_1 + b_2 x_2 + \cdots + b_p x_p + \varepsilon$$

其中 $\varepsilon \sim N(0, \sigma^2)$。令 $\beta = (b_1, b_2, \cdots, b_p)'$，$x = (x_1, x_2, \cdots, x_p)$，上式表示为

$$y = x\beta + \varepsilon$$

现获得 y 和 $x_1, x_2, \cdots, x_p$ 的 n 组观察值(当回归模型中考虑常数项，等价于 x_1 取常数 1)，要求 β 的估计值。设 Y 和 X 分别为相应观察值的 $n \times 1$ 和 $n \times p$ 矩阵，则 β 的估计值

$$\hat{\beta} = (X'X)^{-1} XY$$

对于任意的 y 和 $x_1, x_2, \cdots, x_p$ 的 n 组观察值，均可得到回归系数 β。要判断这一模型的有效性，还要通过对于残差 $r = Y - X\hat{\beta}$ 的分析以检验

$$H_0: \beta = 0,\ H_1: \beta \neq 0$$

stats 函数 regress 给出线性回归方法。使用格式

```
[b, bint, r, rint, stats] = regress(y, x, alpha)
```

其中

y：y 的数据 $n \times 1$ 向量；

x：x 的数据 $n \times p$ 矩阵；

b：β 的估计值；

bint：b 的置信区间；

r：残差 $r = Y - X\hat{\beta}$；

rint：r 的置信区间；

stats：1×3 检验统计量，第一值是回归方程的置信度，第二值是 F 统计量值，第三值是与 F 统计量相应的 p 值，p 很小说明回归方程系数不为 0。

可使用 rcoplot(r, rint)作出残差图。

例如，若 $y = 2 + 0.5x + \varepsilon$，$\varepsilon \sim N(0, 0.1^2)$

```
》x = rand(10, 1);
》x = [ones(10, 1), x];
》y = x*[2; 0.5] + 0.1*randn(10, 1);
》[b, bint, r, rint, stats] = regress(y, x);
》b, stats
    b=
        2.0575
```

0.4281

stats＝

0.7306　　21.6933　　0.0016

可见系数估计基本正确，stats 第一值是回归方程的置信度，超过 70%，第三值表明当显著性水平大于 0.0016，拒绝“H0：b=0”，即认为回归模型成立。

§7.5 实验例题

例 2 （学生的身高、体重与体育成绩）现在我们回头解决引例中的三个问题，如图 7-4 所示。MATLAB 程序文件是下列 eg7_2.m

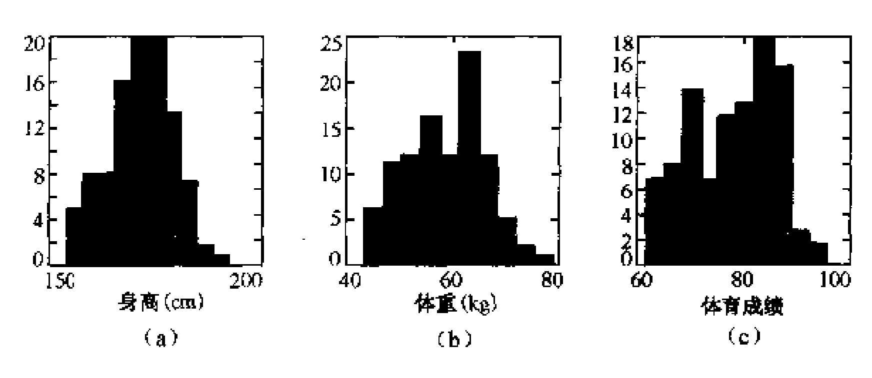


图 7-4 学生数据直方图

(1) 我们利用 hist 画直方图，基本可认为服从正态分布。并计算得

```
datamean =              %身高、体重、成绩均值
  170.1100      59.2300      78.3500
datastd =               %身高、体重、成绩标准差
    6.0834      7.2235      8.1679
```

(2) 是一个参数估计问题，利用 normfit 得到

```
h_hat=                  %身高估计值
    170.1100
h_ci=                   %身高 95%置信区间
    168.9029
    171.3171
w_hat=                  %体重估计值
    59.2300
w_ci=                   %体重 95%置信区间
    57.7967
    60.6633
```

(3) 这是一个假设检验问题，对于身高

$$H0: h=168.3 \quad H1: h\neq 168.3$$

对于体重

$$H0: w=56.2 \quad H1: w\neq 56.2$$

总体方差未知但 $n=100$ 相当大，用 ztest 检验。显著性水平 0.05。

```
h_reply=              %拒绝 H0,即身高有显著区别
    1
h_sig=                %值很小,结论有把握
    0.0029
w_reply=              %拒绝 H0,即体重有显著区别
    1
w_sig=
    6.3338e-007       %值很小,结论有把握
```

(4) 我们从两个方面来看，首先计算相关系数得

```
Cor=
     1.0000    0.8492    0.1834
     0.8492    1.0000    0.1750
     0.1834    0.1750    1.0000
```

可见身高与体重相关性明显(相关系数 0.85)，而成绩与身高或体重相关系数都很小。说明身高成体重对体育成绩几乎没有影响。这与散点图分析吻合。如图 7-5 所示。

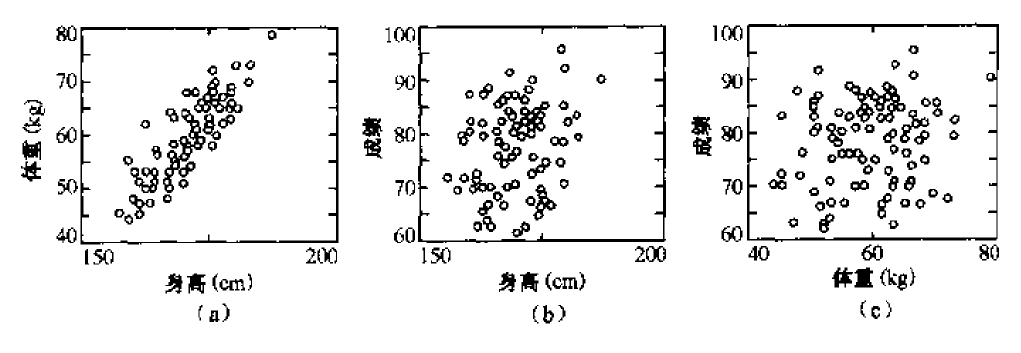


图 7-5 学生数据相关图

若用 regress 建立线性回归模型，得 $s=45.2+0.1676h+0.0779w$

```
b =
   45.2152
    0.1676
    0.0779
stats =
    0.0350    1.7569    0.1780
```

但 stats 的值表明在显著水平 0.05 下模型不成立。残差面有 4 个异常点。如图 7-6 所示。

%M 文件 eg7_2.m

```
clear; close;
data=[167  50  85
179  63  93
……(数据省略)
166  50  69];
```

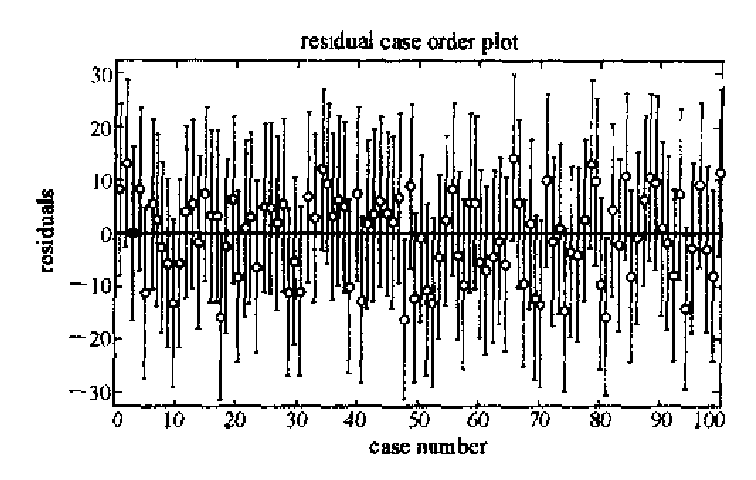


图 7-6 回归分析残差图

```
% 均值和标准差
datamean = mean(data)
datastd = std(data)
% 作图
h = data(:, 1); w = data(:, 2); s = data(:, 3);
subplot(1, 3, 1); hist(h); title('身高');
subplot(1, 3, 2); hist(w); title('体重');
subplot(1, 3, 3); hist(s); title('体育成绩');
% 参数估计
[h_hat, temp, h_ci] = normfit(h, 0.05);
h_hat, h_ci
[w_hat, temp, w_ci] = normfit(w, 0.05);
w_hat, w_ci
% 假设检验
[h_reply, h_sig] = ztest(h, 168.3, std(h), 0.05)
[w_reply, w_sig] = ztest(w, 56.2, std(h), 0.05)
% 相关性分析
Cor = corrcoef(data)
figure;
subplot(1, 3, 1); plot(h, w, 'o'); title('身高—体重');
subplot(1, 3, 2); plot(h, s, 'o'); title('身高—成绩');
subplot(1, 3, 3); plot(w, s, 'o'); title('体重—成绩');
% 回归分析
x = [ones(100, 1), h, w];
[b, bint, r, rint, stats] = regress(s, x, 0.05);
b, stats
figure; rcoplot(r, rint)
```

§7.6 实验习题

1 在上海市环境卫生管区，操作工程师是在有竞争性的公务员考试基础上被录用的。1998年有223人申请15份工作，考试分数排列在下面，将其按增加的顺序排列并作出数据的直方图。在此基础上，有人指控评分是在操纵下进行的，你能解释为什么吗？

34 35 84 64 67 74 43 58 60 90 92 81
78 47 84 69 80 53 66 39 39 37 45 75
47 93 51 44 29 48 83 57 33 45 80 61
48 66 63 44 93 43 43 26 37 33 39 39
30 69 45 76 47 45 43 60 54 45 52 59
42 58 53 62 71 43 42 83 80 74 51 31
50 47 36 56 48 84 58 93 33 53 52 73
90 69 53 39 69 42 52 44 57 36 43 52
67 61 32 30 82 52 30 60 81 56 55 32
53 81 49 91 65 54 31 56 51 68 82 75
58 48 51 92 71 58 84 27 69 59 48 49
91 84 61 68 48 51 47 44 57 95 95 34
69 39 34 46 45 58 69 56 46 58 54 54
92 37 84 37 74 42 60 83 62 46 46 47
48 61 61 37 69 55 80 36 54 91 50 67
61 46 63 48 39 35 76 42 67 43 46 33
49 72 60 44 44 37 58 83 37 60 90 49
27 31 27 66 27 59 56 55 41 40 59 93
43 57 84 45 30 62 33

2 表7-2给出了1930年各国人均年消耗的烟支数以及1950年男子死于肺癌的死亡率（注：研究男子的肺癌死亡率是因为在1930年左右极少的妇女吸烟，记录1950年的肺癌死亡率是因为考虑到吸烟的效应要有一段时间才能显现）。

表 7-2

国家	1930年人均烟消耗量	1950年每百万男子死于肺癌人数	国家	1930年人均烟消耗量	1950年每百万男子死于肺癌人数
澳大利亚	480	180	冰岛	230	60
加拿大	500	150	挪威	250	90
丹麦	380	170	瑞典	300	110
芬兰	1100	350	瑞士	510	250
英国	1100	460	美国	1300	200
荷兰	490	240			

(1) 画出该数据散点图；

(2) 该散点图是否表明在吸烟多的人中间肺癌死亡率较高？

(3) 计算两列数据的相关系数。

3 下图中的 6 个散点图分别具有如下相关系数

-0.85，-0.38，-1.00，0.06，0.60，0.97

请将相关系数与散点图相配。

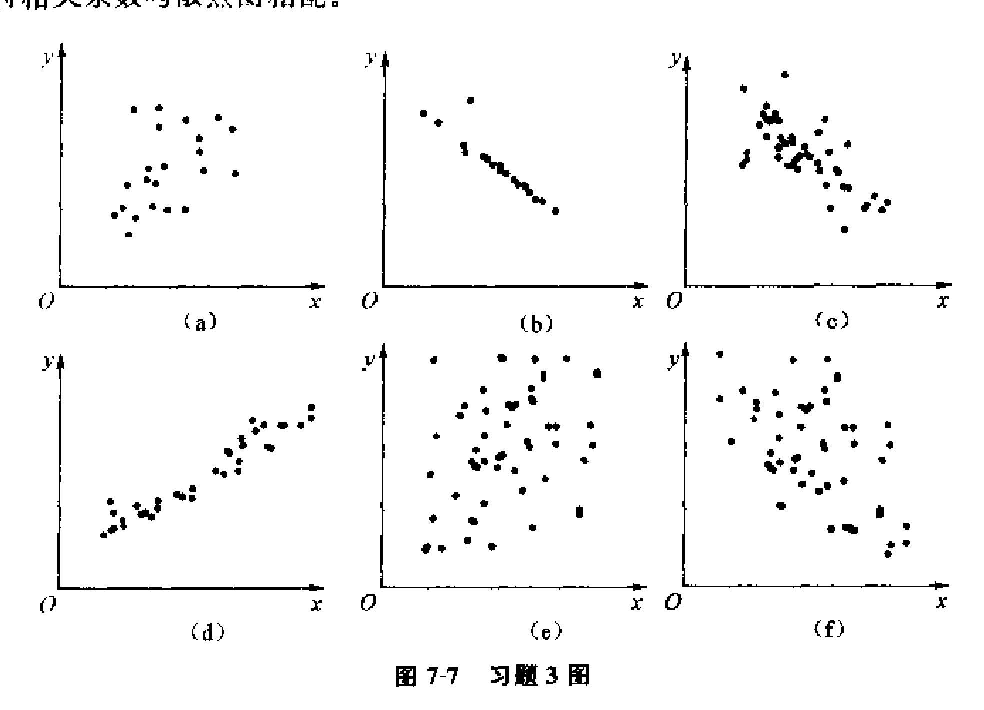


图 7-7 习题 3 图

4 (两个总体检验)设 $x_1, x_2, \cdots, x_m$ 为来自正态总体 ξ(均值 μ_1，方差 σ_1^2)的样本，$y_1, y_2, \cdots, y_n$ 为来自正态总体 η 的样本(均值 μ_2，方差 σ_2^2)(μ_1，μ_2，σ_1，σ_2 已知)，且相互独立，m，n 足够大。

$$H_0: \mu_1 = \mu_2,\ H_1: \mu_1 \neq \mu_2 (\text{或 } \mu_1 > \mu_2,\ \mu_1 < \mu_2)$$

检验统计量 $U = \dfrac{\bar{x} - \bar{y}}{\sqrt{\dfrac{s_x^2}{m} + \dfrac{s_y^2}{n}}} \sim N(0, 1)$，写出拒绝域并编写 MATLAB 程序。

5 某大学随机地取 200 名男生和 100 名女生，查阅他们的微积分期终成绩，结果发现男生的平均分为 73 分，标准差为 17 分；女生的平均分为 76 分，标准差为 15 分，所观察到的数据能否断言女生的数学成绩优于男生？如果标准差分别为 7 分和 5 分呢？

6 某保健食品商声称学生服用该保健食品一个月后能提高他们的数学能力和成绩，为了查明此保健食品是否真的那么神，设计了一次实验：随机地选取 500 名学生，并将他们随机地分为两个组，甲组服用保健食品，乙组服用模样、品味与保健食品一样的葡萄糖丸。两组同学以为自己在服用保健食品，一个月后进行一次数学考试，结果甲组的平均分是 73 分，标准差为 18 分；乙组的平均分是 71 分，标准差为 17 分。其间的差异是由于机会变异引起还是保健食品真的起了作用？

7 20 世纪 60 年代，美国提出了一种“负收入税”，对低收入进行补助而不是向他们收税，这项福利计划会造成受益人不工作吗？在新泽西州的三个城市做了一项实验来寻求答案。母体由这些城市的 1 万个低收入家庭组成，从中随机选出 225 个家庭实施负收入税，400 个家庭作为对照，假定这 400 个家庭不会因为他人获补助而对自身产生影响。对这 625 个家庭跟踪 3 年。

(1) 对照组家庭在 3 年期间平均受雇工作 7000 小时，标准差是 3900 小时；获益家庭则平均受雇工作 6200 小时，标准差是 3400 小时，两个平均数之间的差是属于机会变异吗？

(2) 对照组中，88%的户主受雇，而实施负收入税的家庭中，82%的户主受雇，这两上百分数之间的差异是统计显著的吗？

8 得到某商品的需求量、消费者的平均收入、商品价格的统计数据如表 7-3 所示，建立回归模型并进行检验，估计平均收入为 1000，价格为 6 时的商品需求量。

表 7-3

需求量	100	75	80	70	50	65	90	100	110	60
收　入	1000	600	1200	500	300	400	1300	1100	1300	300
价　格	5	7	6	6	8	7	5	4	3	9

9 某人记录了 21 天使用空调器的时间和使用烘干器的次数，并监测电表以计算出每天的耗电量(kWh)与空调器使用的小时数和烘干器使用次数之间的关系，记录数据见表 7-4，试建立并检验回归模型，诊断是否有异常点。

表 7-4

序　号	1	2	3	4	5	6	7	8	9	10	11
耗电量(kWh)	35	63	66	17	94	79	93	66	94	82	78
小时数	1.5	4.5	5.0	2.0	8.5	6.0	13.5	8.0	12.5	7.5	6.5
使用次数	1	2	2	0	3	3	1	1	1	2	3

序　号	12	13	14	15	16	17	18	19	20	21
耗电量(kWh)	65	77	75	62	85	43	57	33	65	33
小时数	8.0	7.5	8.0	7.5	12.0	6.0	2.5	5.0	7.5	6.0
使用次数	1	2	2	1	1	0	3	0	1	0

§7.7 补充知识：非线性回归

1. 坏数据去除

若确认列状数据为来自同一总体的样本，那么绝大部分数据应在均值的 4 或 5 个标准偏差之内，因一些外在因素往往会有一些“坏数据”(异常大或异常小)混杂其中，这些“坏数据”往往会严重影响统计量的计算结果，从而影响统计推断的正确性，所以应该去除。另一种方式是按一定百分比剔除掉最大和最小的一部分数据，例如各剔除 2.5%。坏数据去除后，统计量需重新计算。MATLAB 统计数据箱函数 trimmean 提供了后一种方式。

trimmean(data, percent)　忽略数据上下各 $\frac{\text{percent}}{2}$%后的均值。0<=percent<=100。

设列状数据为 cdata，可用下列 M 函数 trim.m 去除“坏数据”(默认按 4 倍标准偏差)

```
        %M 函数 trim.m
function data = trim(data, outval)
```

```
% 去除坏数据,包括 NaN, Inf 和异常大小数据
% data:列状数据,每列来自一个总体
% outval:系数因子,离均值超过 outval 倍标准差被判为异常大小,默认值为 4
if nargin<2,outval = 4; end
outliers = (isnan(data)|abs(data) == inf);
[n, m] = size(data); mu = mean(data); sigma = std(data);
outliers = outliers + (abs(data -
ones(n, 1)* mu))outval* ones(n, 1)* sigma);
if m>1,
    data(any(outliers'),:) = [];
else
    data(find(outliers'),:) = [];
end
```

然后在命令窗口用

```
》clear; c = [ones(100, 1);100];        %含有一个不正常的大数 100
》mean(c), trimmean(c, 5)                %trimmean 忽略上下各 2.5%的均值
ans =
   1.9802
ans =
   1
》c = trim(c); mean(c)                   %坏数据已被清除

ans=
  1
```

2. 单个正态总体期望的 t 检验(小样本检验)

若 n 不够大且方差 σ^2 未知,z 检验 ztest 不能用。对于正态总体,可用 t 检验。检验统计量 $T=\dfrac{\overline{x}-\mu_0}{s/\sqrt{n}}\sim t(n-1)$,拒绝域 $|T|>t_{1-\alpha/2}$(或 $T<t_\alpha$, $T<t_{1-\alpha}$)。stats 函数 ttest 给出 t 检验法。使用格式

```
[h, sig] = ttest(x, m, alpha, tail)
```

参数含义同 ztest。

3. 两个正态总体期望的 t 检验(小样本检验)

设 $x_1, x_2, \cdots, x_m$ 为来自正态总体 $N(\mu_1, \sigma^2)$ 样本,$y_1, y_2, \cdots, y_n$ 为来自正态总体 $N(\mu_2, \sigma^2)$ 样本(μ_1, μ_2, σ 未知),且相互独立。

$$H_0:\mu_1=\mu_2,\ H_1:\mu_1\neq\mu_2(\text{或 }\mu_1>\mu_2,\ \mu_1<\mu_2)$$

检验统计量 $T=\dfrac{\overline{x}-\overline{y}}{s_w\sqrt{\dfrac{1}{m}+\dfrac{1}{n}}}\sim t(m+n-2)$，其中 $s_w=\sqrt{\dfrac{(m-1)s_x^2+(n-1)s_y^2}{m+n-2}}$

拒绝域 $|T|>t_{1-\alpha/2}$（或 $T>t_{1-\alpha}$，$T<t_\alpha$）。使用 stats 函数 ttest2。使用格式

```
[h, sig] = ttest2(x, y, alpha, tail)
```

参数含义与 ztest 类似。

例 3　对两种牌子的毛纺织品进行强度试验，已经测得数据（磅/平方英寸）见表 7-5。

表　7-5

甲　牌	138	127	134	125		
乙　牌	154	157	135	140	130	134

设毛纺织品强度服从正态分布，取 $\alpha=0.05$，问两个牌子的毛纺织品平均强度是否有显著差异？（设方差相等）

解：本问题是两正态总体期望的假设检验问题。设 μ_1，μ_2 分别为两个牌子的毛纺织品平均强度，

$$H_0:\mu_1=\mu_2,\ H_1:\mu_1\neq\mu_2$$

利用 t 检验 ttest2 解决。

```
》clear; x=[138  127  134  125]; y=[154  157  135  140  130  134];
》[h, sig] = ttest2(x, y)
h=
    0
sig=
    0.1239
```

可见不应拒绝 H_0，即应认为二者无明显差异。

4. 拟合度 χ^2 检验

假定样本 S 有 k 个可能的大体取值或有 k 个分组，实际组频数分别为 x_1，x_2，…，x_k（一般要求大于 3）。现在要分析这批数据与一已知概率分布 P 是否相符。

H_0：S 来自 P，H_1：S 不来自 P

检验统计量 $\chi^2=\sum\limits_{i=1}^{k}\dfrac{(x_i-p_i)^2}{p_i}\sim\chi^2(k-1)$，其中 p_i 是分布 P 相应于 k 个分组的理论组频数。拒绝域 $\chi^2>\chi^2_{1-\alpha}$。下列 M 函数 chi2test.m 给出这一检验法。

```
            %M 函数 chi2test.m
function [h, sig] = chi2test(n, x, p, alpha)
% 分布拟合度检验
% n:数据总数
% x:实际组频数向量
% p:理论组频数向量
```

```
  if nargin<4, alpha = 0.05; end
  if min(size(x))~ = 1|min(size(p))~ = 1,
      error('x, p must be vectors');
  end
  kx = length(x); kp = length(p);
  if kx~ = kp,
      error('Dimension of vectors must be agree');
  end
  chi2 = sum((x-p).^2./p);
  h = 0;
  sig = 1 - chi2cdf(chi2, kx - 1);
  if sig<alpha, h = 1; end
```

作为一个示例，生成 100 个正态分布随机数，再检验这批数据是否服从正态分布。

```
》clear; close;
》data = normrnd(600, 163, 100, 1);      %模拟数据
》hist(data, 10)
》[n, x] = hist(data, 10);      % n 为实际组频数向量
》xbar = mean(data), sigma = std(data)
xbar=
  610.9822
sigma=
  163.2051
》g = (x(1:(length(n) - 1)) + x(2:length(n)))/2;
》g1 = [0, g]; g2 = [g, 1300];
》p = normcdf(g2, xbar, sigma) - normcdf(g1, xbar, sigma);
》p = 100* p;[h, sig] = chi2test(100, n, p)      % p 为理论组频数向量
h=
    0                                    %接受原假设
sig=
    0.4402
```

5. 独立性 χ^2 检验

设变量 ξ 的取值分为 s 组，η 的取值分为 t 组，C_{ij} 表示 ξ 取第 i 组、η 取第 j 组的频数。

H_0：ξ 与 η 相互独立，H_1：ξ 与 η 不独立

检验统计量 $\chi^2 = \sum_{i=1}^{s}\sum_{j=1}^{t}\frac{(C_{ij}-E_{ij})^2}{E_{ij}} \sim \chi^2(st)$，其中 $E_{ij} = \frac{\sum_{k=1}^{t}C_{ik}\sum_{k=1}^{s}C_{kj}}{\sum_{i=1}^{s}\sum_{j=1}^{t}C_{ij}}$。

拒绝域 $\chi^2 > \chi^2_{1-\alpha}$。

例 4　研究人员试图评价某拥有 6000 多雇员的保健组织内的禁烟政策。在禁烟令设

立后4个月，研究人员随机地选择687名雇员进行民意调查。他们给这些雇员中的每一位寄去了问答卷并要求匿名返回。454位寄回了问答卷。回答者的年龄、性别和受雇时间的分布与整体雇员的相类同。回答者提供了吸烟状况和对禁烟令的认可的信息，见表7-6。

表 7-6

吸烟状况	对禁烟令的态度		
	认　可	不认可	不确定
从未吸烟	237	3	10
以前吸烟者	106	4	7
现时吸烟者	24	32	11

"吸烟状况"与"对禁烟令的态度"是雇员总体的两个变量，等价地，它们相互独立吗？

解：本题为独立性 χ^2 检验，设

H_0："吸烟状况"与"对禁烟令的态度"独立

H_1："吸烟状况"与"对禁烟令的态度"不独立

写出下列M脚本eg7_4.m。

M脚本 eg7_4.m

```
clear; C=[237  3  10;  106  4  7;  24  32  11];
E=sum(C')'*sum(C)/sum(sum(C));
chi2=sum(sum((C-E).^2./E))
crit=chi2inv(0.95, 9)
h=chi2>crit
```

计算结果h=1，拒绝H_0，即认为不独立。

```
chi2=
    168.1112
crit=
    16.919
h=
    1
```

6. 非线性回归

[beta, R, J]=nlinfit(x, y, 'model', beta0)

其中：model：模型的M函数名，此M函数形式为

y=f(beta, x)，beta为参数；

x：因素数据矩阵，每列一个变量；

y：响应数据向量；

beta0：参数迭代新值；

beta：返回参数估计值；

R：返回残差；

J：返回用于估计预测误差的Jacobi矩阵。

例 5 (化学反应速度与反应物含量)在研究化学反应过程中,建立了一个反应速度和反应物含量的数学模型,形式为

$$y=\frac{\beta_1 x_2-\frac{x_3}{\beta_5}}{1+\beta_2 x_1+\beta_3 x_2+\beta_4 x_3}$$

其中 β_1, β_2, β_3, β_4, β_5 是未知的参数,x_1, x_2, x_3 是三种反应物(氢、N 戊烷、异构戊烷)的含量,y 是反应速度。今测得一组数据见表 7-7,试由此确定参数 β_1, β_2, β_3, β_4, β_5。已给其参考值为(1, 0.05, 0.02, 0.1, 2)。

表 7-7 化学反应速度与反应物含量

序号	反应速度 y	氢 x_1	N 戊烷 x_2	异构戊烷 x_3
1	8.55	470	300	10
2	3.79	285	80	10
3	4.82	470	300	120
4	0.02	470	80	120
5	2.75	470	80	10
6	14.39	100	190	10
7	2.54	100	80	65
8	4.35	470	190	65
9	13.00	100	300	54
10	8.5	100	300	10
11	0.05	100	80	120
12	11.32	285	300	10
13	3.13	285	190	120

%M 函数 statsfun.m

```
function y = statsfun(beta, x)
b1 = beta(1);
b2 = beta(2);
b3 = beta(3);
b4 = beta(4);
b5 = beta(5);
x1 = x(:, 1);
x2 = x(:, 2);
x3 = x(:, 3);
y = (b1*x2 - x3/b5)./(1 + b2*x1 + b3*x2 + b4*x3);
```

求解如下

```
»x = [470, 285, 470, 470, 470, 100, 100, 470, 100, 100, 100, 285, 285;
```

```
    300, 80, 300, 80, 80, 190, 80, 190, 300, 300, 80, 300, 190;
    10, 10, 120, 120, 10, 10, 65, 65, 54, 120, 120, 10, 120]';
》y = [8.55, 3.79, 4.82, 0.02, 2.75, 14.39, 2.54, 4.35, 13, ...
    8.5, 0.05, 11.32, 3.13];
》beta0 = [1, 0.05, 0.02, 0.1, 2]';
》[beta, R, J] = nlinfit(x, y, 'statsfun', beta0);
》beta                    %回归系数
 beta=
          1.2526
          0.0628
          0.0400
          0.1124
          1.1914
》betaci = nlparci(beta, R, J)      %置信区间
    betaci=
        -0.7467    3.2519
        -0.0377    0.1632
        -0.0312    0.1113
        -0.0609    0.2857
        -0.7381    3.1208
》[ypre, delta] = nlpredci('statsfun', x, beta, R, J);      %预测(估计)
》plot(x(:, 1), y,'o', x(:, 1), ypre, '*');
```

结果如图 7-8 所示。

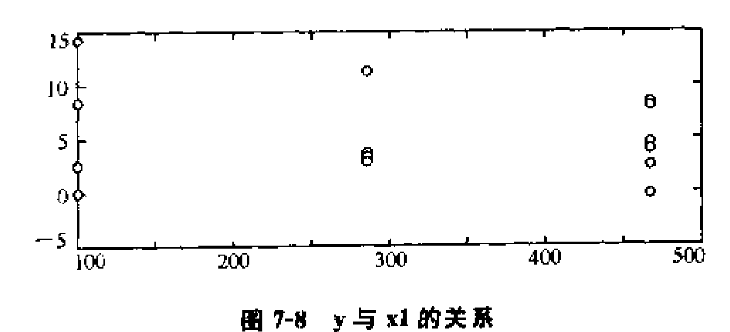


图 7-8　y 与 x1 的关系

可见从第一个变量关系上用回归方程得出的估计值与真值很接近。

7. 补充习题

10　学生中流传在衣阿华州立大学黄道十二宫上行走将会在下一次考试中失败这一荒诞说法，随机抽取 458 名学生询问他们的忌讳情况：195 名在黄道上行走而 263 名选择绕过它。在黄道十二宫上行走的 195 名学生中 60 名女生，另 135 名是男生。避开黄道的 110 名是女生，153 名是男生。在黄道上行走的行为这一属性与性别属性是否存在着相关性？

11　怀孕妇女分娩开始时间在一天 24 小时内是一致的吗？为揭示该问题，研究人员记录了 1186 名怀孕妇女的分娩时间，他们将从半夜开始共 24 个小时的观察值列在表 7-8 中。

数据是否表明分娩开始时间在一天 24 小时内一致?

表 7-8

小 时	频 数	小 时	频 数	小 时	频 数	小 时	频 数
1	52	7	58	13	21	19	47
2	73	8	47	14	31	20	34
3	89	9	48	15	40	21	36
4	88	10	53	16	24	22	44
5	68	11	47	17	37	23	78
6	47	12	34	18	31	24	59

12 (商品销售量与价格)某厂生产的一种电器的销售指数 Y 主要与竞争对手的价格 X_1 和本厂的价格 X_2 有关。表 7-9 是该商品在 10 个城市的销售记录,试根据这些数据建立 Y 与 X_1 和 X_2 的关系式,对得到的模型和系数进行检验。若某市本厂产品售价 160(元),竞争对手售价 170(元),预测商品在该市的销售指数。

表 7-9 商品销售量 Y 与价格 X_1 和 X_2

X_1(元)	120	140	190	130	155	175	125	145	180	150
X_2(元)	100	110	90	150	210	150	250	270	300	250
Y	102	100	120	77	46	93	26	69	65	85

实验八　凸轮设计　插值与拟合

本实验中我们学习数据插值和拟合的基本方法和相关的 MATLAB 命令，并研究凸轮设计和人口数据拟合等应用问题。补充知识介绍了 MATLAB 的样条函数工具箱和多元插值方法以及海底测量问题不规则数据插值。

§8.1　引例：万能拉拔机凸轮设计

在万能拉拔机中有一个圆柱形凸轮，其底圆半径 $R = 300$mm，凸轮的上端面不在同一平面上（如图 8-1 所示），而要根据从动杆位移变化的需要进行设计制造。

根据设计要求，将底圆周 18 等分，旋转一周。第 i 个分点对应柱高 y_i（$i=0, 1, 2, \cdots, 18$），拔据见表 8-1。为了数控加工，需要计算出圆周任一点的柱高。

表 8-1　凸轮高度的数据（单位：mm）

i	0 和 18	1	2	3	4	5
y_i	502.8	525.0	514.3	451.0	326.5	188.6
i	6	7	8	9	10	11
y_i	92.2	59.6	62.2	102.7	147.1	191.6
i	12	13	14	15	16	17
y_i	236.0	280.5	324.9	369.4	413.8	458.3

我们将圆周展开，画出对应的柱高曲线。

```
》clear; close;
》x = linspace(0, 2*pi*300, 19);
》y = [502.8  525.0  514.3  451.0  326.5  188.6  92.2  59.6  62.2  102.7...
    147.1  191.6  236.0  280.5  324.9  369.4  413.8  458.3  502.8];
》plot(x, y, 'o'); axis([0, 2000, 0, 550]);
```

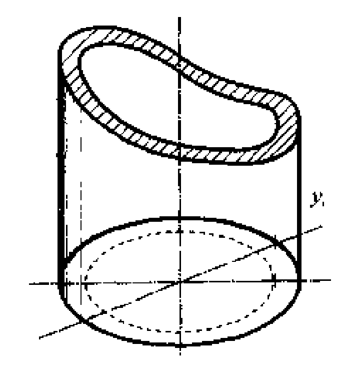


图 8-1　万能拉拔机凸轮示意图

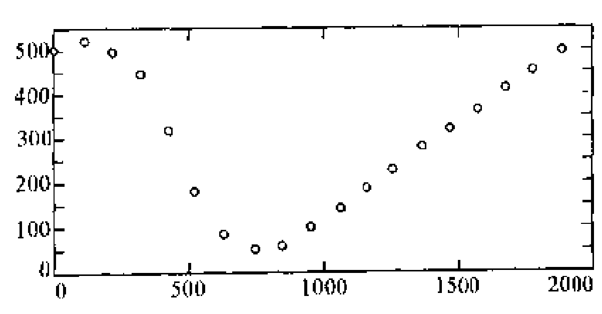


图 8-2　凸轮柱高数据图

可见柱高形成一条V形曲线。如图8-2所示。现在的问题是，怎样给出分点之外的柱高呢？

§8.2 理论基础：数据插值和拟合

在生产实践和科学研究中，常常有这样的问题：由实验或测量得到变量间的一批离散样点，要求得到变量之间的函数关系或得到样点之外的数据。与此有关的一类问题是**插值**问题。当原始数据(x_0, y_0)，(x_1, y_1)，…，(x_n, y_n)精度较高，要求确定一个初等函数$y = P(x)$（一般用多项式或分段多项式函数）通过已知各数据点，即

$$y_i = P(x_i), \ i = 0, 1, \cdots, n$$

另一类是**数据拟合**问题。当我们已经有了函数关系式，而其中参数未知或原始数据有误差时，我们确定的初等函数$y = P(x)$并不要求经过数据点，而是要求在某种距离意义下的误差达到最小（通常考虑使各数据点误差平方和最小）。

1. 分段线性插值

这是最通俗的一种方法，直观上就是将各数据点用折线连接起来。如果

$$a = x_0 < x_1 < \cdots < x_n = b \tag{8.1}$$

那么分段线性插值公式为

$$P(x) = \frac{x - x_i}{x_{i-1} - x_i} y_{i-1} + \frac{x - x_{i-1}}{x_i - x_{i-1}} y_i, \ x_{i-1} < x \leqslant x_i, \ i = 1, 2, \cdots, n \tag{8.2}$$

可以证明，当分点足够细，分段线性插值是收敛的。其缺点是不能形成一条光滑曲线。

2. 多项式插值

给定在$x_0, x_1, \cdots, x_n$点的值为$y_0, y_1, \cdots, y_n$，设有m次多项式

$$P(x) = a_0 x^m + a_1 x^{m-1} + \cdots + a_{m-1} x + a_m$$

通过所有$n + 1$个点。那么

$$a_0 x_i^m + a_1 x_i^{m-1} + \cdots + a_{m-1} x_i + a_m = y_i, \ i = 0, 1, \cdots, n \tag{8.3}$$

可以证明当$m = n$且$x_0, x_1, \cdots, x_n$互不相同，这样的多项式存在且唯一。若要求得函数表达式，可直接解方程组(8.3)。若只要求得函数在插值点处散值，可利用下列**Lagrange插值公式**

$$P_n(x) = \sum_{i=0}^{n} y_i \left(\prod_{j \neq i, j=0}^{n} \frac{x - x_j}{x_i - x_j} \right) \tag{8.4}$$

多项式插值光滑但不具有收敛性，一般不宜采用高次多项式（如$m > 7$）插值。

3. 样条插值

样条本来是绘图员用于数据放样的工具。在画曲线时要求经过一些设定值且值整条曲线都很光滑。以后逐渐发展成为一个应用极为广泛的数学分支。现在数学上所说的样条，实质上指分段多项式的光滑连接。

设有区间$[a, b]$的一个划分(8.1)式，称分段函数$S(x)$为**k次样条函数**，若它满足

(1) $S(x)$在每个小区间上是次数不超过k次的多项式；

(2) $S(x)$在$[a, b]$上具有直到 $k-1$ 阶连续导数；

用样条函数作出的插值称为**样条插值**。工程上广泛采用三次样条插值。

n 段三次多项式共有 $4n$ 个参数，光滑性条件含 $3(n-1)$ 个约束，插值条件含 $n+1$ 个约束，从而三次样条插值结果不唯一。另外需要两个定解条件。通常有下列 4 类条件：

(1) **非扭结**：第一、二端多项式三次项系数相同，最后一段和倒数第二段三次项系数相同；

(2) **一阶导数**：$S'(x_0) = y_0'$，$S'(x_n) = y_n'$；

(3) **二阶导数**：$S''(x_0) = y_0''$，$S''(x_n) = y_n''$；

特别有**自然样条**：$S''(x_0) = 0$，$S''(x_n) = 0$；

(4) **周期样条**：$S'(x_0) = S'(x_n)$，$S''(x_0) = S''(x_n)$（前提条件 $S(x_0) = S(x_n)$）。

当原函数为周期函数或封闭曲线，宜使用周期样条。

理论上，插值样条函数可由约束条件方程组解出，但这一方程组往往是病态的。克服这一困难的方法有两种。一类是"三弯距"法，将问题转化为一个良态方程组求解；另一类是 ***B* 样条**，这一方法与 Lagrange 插值公式类似，可不必解方程组（参见文献[9]）。

4. 最小二乘拟合

假设已知函数 $y = f(c, x)$（这里 c 和 x 均可为向量）的一批有误差的数据 (x_i, y_i)，$i = 0, 1, \cdots, n$，要求据此确定参数 c。这样的问题称为**数据拟合**。**最小二乘法**就是求 c 使得残差平方和

$$Q(c) = \sum_{i=0}^{n}(y_i - f(c, x_i))^2 \tag{8.5}$$

达到最小。当 f 关于 c 是线性函数，问题转化为一个线性方程组求解，且其解存在唯一。如果 f 关于 c 是非线性函数，问题转化为一个函数极值问题，其解往往不唯一且依赖于初值选取。这里的建模原理实质上与实验七中回归分析是一致的。

§8.3 数据拟合 MATLAB 命令

polyfit	多项式拟合和插值；	leastsq	最小二乘法；
polyval	多项式求值；	curvefit	曲线拟合；
interp1	一元插值；	csape	各种边界条件的样条插值；
spline	样条插值；	fnplt	样条结构的图形；
ppval	pp 样条求值；	csaps	样条光滑拟合；
unmkpp	pp 样条展开；	interp2	二元插值；
mkpp	形成 pp 式；	griddata	杂乱数据插值。

leastsq，curvefit 需要优化工具箱（optim）支持，csape，fnplt，csaps 需要样条函数工具箱（splines）支持。

1. 多项式插值和拟合

对于多项式函数的插值和拟合，有一个方便的方法。

> p = polyfit(x, y, k)　用 k 次多项式拟合向量数据(x, y)，返回多项式的降幂系数。当 k≥n－1 时，polyfit 实现多项式插值。

例如，现有数据。

x	0.1	0.2	0.15	0	−0.2	0.3
y	0.95	0.84	0.86	1.06	1.50	0.72

```
》clear; x=[0.1, 0.2, 0.15, 0, -0.2, 0.3];
》y=[0.95, 0.84, 0.86, 1.06, 1.50, 0.72];
》p=polyfit(x, y, 2)        %二次拟合多项式 p(1)x²+p(2)x+p(3)
  p=
      1.7432    -1.6959    1.0850
》xi=-0.2:0.01:0.3;
》yi=polyval(p, xi); subplot(2, 2, 1);
》plot(x, y, 'o', xi, yi, 'k');
》title('polyfit');
》p=polyfit(x, y, 5)        %五次拟合多项式(等价于多项式插值)
  p=
      1.0e+003
      -1.8524    0.7560    0.0079    -0.0275    0.0010    0.0011
》yi=polyval(p, xi); subplot(2, 2, 2);
》plot(x, y, 'o', xi, yi, 'k');
》title('polyinterp');
```

2. 一元插值

```
yi=interp1(x, y, xi)  根据数据(x, y)给出在 xi 的分段线性插值结果 yi。
yi=interp1(x, y, xi, 'spline')  使用三次样条插值。
yi=interp1(x, y, xi, 'cubic')  使用分段三次插值。
```

interp1 要求 x 是单调上升的，且只能做内插。对于上述数据，改写

```
》x=[-0.2, 0, 0.1, 0.15, 0.2, 0.3];  %单调上升
》y=[1.50, 1.06, 0.95, 0.86, 0.84, 0.72];  %当然 y 也相应改变
》xi=-0.2:0.01:0.3;
》yi=interp1(x, y, xi);           %分段线性插值
》subplot(2, 2, 3)
》plot(x, y, 'o', xi, yi, 'k')
》title('linear');
》yi=interp1(x, y, xi, 'spline');   %三次样条插值
》subplot(2, 2, 4)
》plot(x, y, 'o', xi, yi, 'k')
》title('spline');
```

直观上，多项式插值误差大，多项式拟合不过数据点，分段线性插值不光滑，样条插值过数据

点且光滑(如图 8-3 所示)。

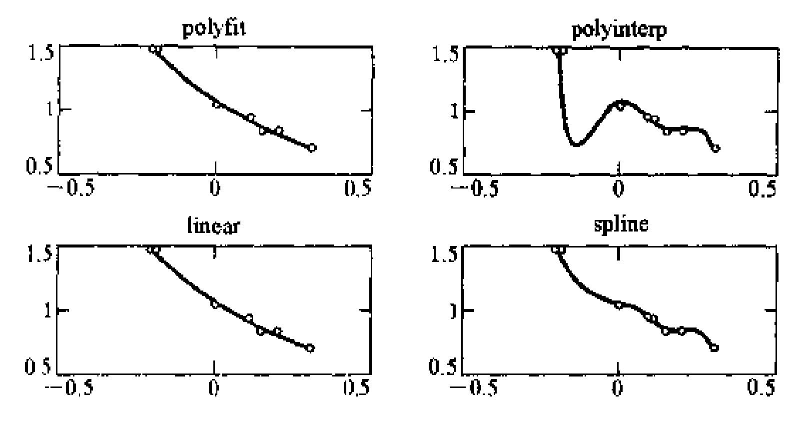


图 8-3 各种插值结果

3. 三次样条插值

YI = spline(x, y, xi) 等价于 YI = interp(x, y, xi, 'spline');
pp = spline(x, y) 返回样条插值的分段多项式(pp 形式);
spline 要求 x 是单调上升的,且只能做内插。
[breaks, coefs] = unmkpp(pp) 将 pp 形式展开,其中 breaks 为结点,coefs 为各段多项式系数。
yi = ppval(pp, xi), pp 形式在 xi 的函数值。

考虑上述数据

```
》pp = spline(x, y);
》[b, c] = unmkpp(pp)
b=
    -0.2000         0        0.1000      0.1500      0.2000      0.3000
c =
   -36.3850    21.8592    -5.1164      1.5000
   -36.3850     0.0282    -0.7390      1.0600
   227.6995   -10.8873    -1.8249      0.9500
  -143.0047    23.2676    -1.2059      0.8600
  -143.0047     1.8169     0.0484      0.8400
```

b 显示样条函数的 6 个结点,c 显示样条函数的五个分段三次多项式,其中第一段为

$$-36.3850(x+0.2)^3+21.8592(x+0.2)^2-5.1164(x+0.2)+1.5000,\ 0.2\leqslant x\leqslant 0$$

spline 使用"非扭结"端点条件,即强迫第一、二端多项式三次项系数相同,最后一段和倒数第二段三次项系数相同。这里若用

```
》yi = ppval(pp, xi)
```

所得结果应与上述 yi = interp1(x, y, xi, 'spline')一致。

4. 非线性最小二乘拟合

c = leastsq('fun', c0) 使用迭代法搜索最优参数 c。其中自变量为参数 c 的函数 fun 是误差向量 y−f(c, x) (x, y 为数据);
c = curvefit('fun2', c0, x, y) 使用更方便,从外部输入数据,这里 fun2 为函数 f(c, x)。

leastsq, curvefit 是优化工具箱(optim)函数,求解一般最小二乘曲线拟合问题。设有函数 $y = f(c, x)$,其中 c 为参数向量,现有一批有误差的数据。

x	x_1	x_2	…	x_n
y	y_1	y_2	…	y_n

可用 leastsq 或 curvefit 求解。考虑用指数函数 $y = ae^{bx}$ 拟合上述数据。

```
%M 函数 fit.m
function  e = fit(c)
x = [0.1  0.2  0.15  0  -0.2  0.3];
y = [0.95  0.84  0.86  1.06  1.50  0.72];
e = y - c(1)* exp(c(2)* x);
```

然后在命令窗口执行

```
》c = leastsq('fit', [1, 1])     %这里[1, 1]为迭代初值
c=
    1.0998    -1.4923     %结果函数 1.0998* exp(-1.4923x)
```

若用 curvefit, 先写 M 函数。

```
%M 函数 fit2.m
function  f = fitfun2(c, x)
f = c(1)* exp(c(2)* x);
》x = [0.1, 0.2, 0.15, 0, -0.2, 0.3];
》y = [0.95, 0.84, 0.86, 1.06, 1.50, 0.72];
》c = curvefit('fit2', [1, 1]', x, y)
c=
    1.0998
   -1.4923              %注意两函数在表达优化函数上的区别
```

最小二乘法能找到符合经验公式的最优曲线,但是这一经验公式是否有效,还需要分析检验。一般可从图象上作出判断,定量方法是计算残差平方和,再进行统计检验(事实上就是用非线性回归分析 nlinfit,参看实验七补充知识)。

5. 线性最小二乘拟合

线性最小二乘拟合可直接用求解超定线性方程组的方法,而有些非线性函数也可化为线性问题求解。例如上述函数 $y = ae^{bx}$ 两边取对数得

$$z = \ln y = \ln a + bx$$

然后通过除法解超定方程组(见实验一) $z_i = \ln a + bx_i$ 得 $\ln a$ 和 b。例如

```
》x = [0.1, 0.2, 0.15, 0, -0.2, 0.3];
```

```
»z = log([0.95, 0.84, 0.86, 1.06, 1.50, 0.72]');
»m = [ones(6, 1),x'];
»c = m\z; a = exp(c(1)), b = c(2)
a=
    1.0969
b=
  -1.4476
```

注意到这里结果与前面有些差异，因为通过对数变换，误差平方和的含义改变了。线性最小二乘拟合适用于多元函数。进一步可用线性回归 regress 分析拟合效果(见实验七)。

§8.4 实验例题

例1 (凸轮设计)由于没有拟合经验公式，又要求严格按设计数据要求(而这些数据应认为是足够验确的)，所以应该用插值方法。下列 M 文件 eg8_1.m 使用了分段线性插值和样条插值两种方法。圆周加工步长为 $h=1\text{mm}$。结果表明分段线性插值不光滑，这样加工的凸轮运行不平稳(图 8-4(a))，而样条插值具有光滑性，所以较好(图 8-4(b))。

```
                %M 文件 eg8_1.m
clear; close;
x = linspace(0, 2*pi*300, 19);
y = [502.8  525.0  514.3  451.0  326.5  188.6  92.2  59.6  62.2  102.7...
   147.1  191.6  236.0  280.5  324.9  369.4  413.8  458.3  502.8];
xi = 0:2*pi*300;
yi = interp1(x, y, xi);
subplot(1, 2, 1); plot(xi, yi); title('Linear');
yi = spline(x, y, xi);
subplot(1, 2, 2); plot(xi, yi); title('Spline');
```

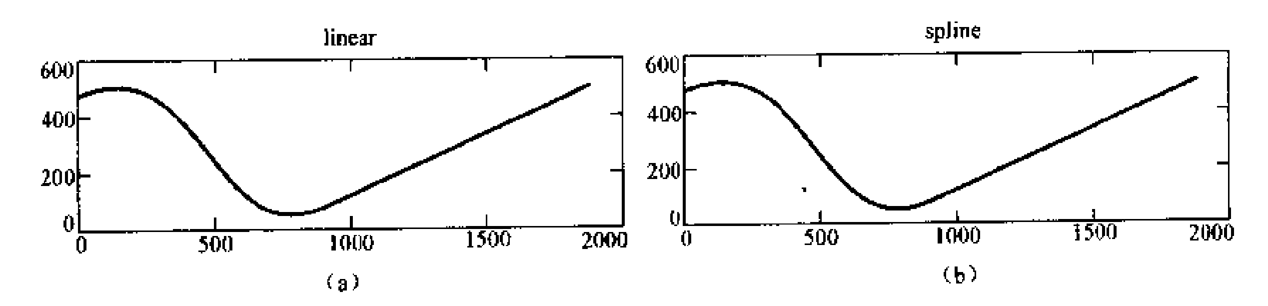


图 8-4 凸轮柱面插值曲线

(a) 线性插值；(b) 样条插值

仔细分析可以发现，上述结果还有一个缺陷。虽然样条插值具有光滑性，但对两头不起作用。当我们把两头合拢，接头并不光滑，原因是 spline 使用非扭结端点条件。这个问题可改用周期性端点条件来克服，见补充知识。

例2 (人口预测)表 8-2 为美国人口两个世纪以来的统计数据，试依此建立美国人口端

长的数学模型，并预测 2010 年和 2020 年的美国人口。

表 8-2

年 人口(百万)	1800 5.3	1810 7.2	1820 9.6	1830 12.9	1840 17.1	1850 23.2	1860 31.4
年 人口(百万)	1870 38.6	1880 50.2	1890 62.9	1900 76.0	1910 92.0	1920 106.5	1930 123.2
年 人口(百万)	1940 131.7	1950 150.7	1960 179.3	1970 204.0	1980 226.5	1990 251.4	2000 275.0

解：我们先作出数据图，可以看见大致接近一指数函数(图 8-5)。

```
》clear; close;
》t = 1800:10:2000;
》N = [5.3  7.2  9.6  12.9  17.1  23.2  31.4 ...
》    38.6  50.2  62.9  76.0  92.0  106.5  123.2 ...
》    131.7  150.7  179.3  204.0  226.5  251.4  275.0];
》plot(t, N, 'o')
```

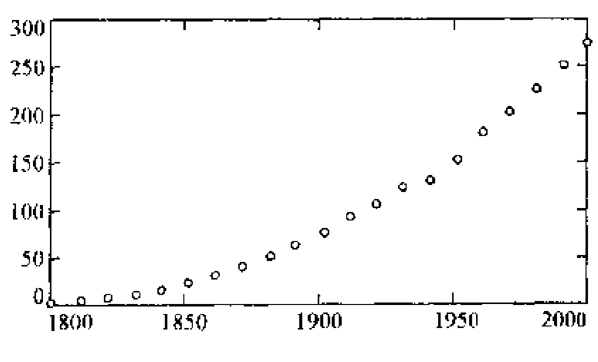


图 8-5 人口数据

我们现在来从机理上分析人口问题数学模型。人口的出生率 b 和死亡率 d 可设为常数，第 t 年人口数为 $N(t)$，那么在一个较小的时间段 $[t, t+\Delta t]$ 内新增人口

$$N(t+\Delta t) - N(t) = (b-d)N(t)\Delta t$$

令 $r = b-d$，$\Delta t \to 0$，得

$$N'(t) = rN(t) \tag{8.6}$$

设 $N(t_0) = N_0$，那么

$$N(t) = N_0 \exp(r(t-t_0)) \tag{8.7}$$

此为人口学 **Malthus 模型**。可见对数据图的推测是有道理的。

最后我们利用历史数据来确定参数 N_0 和 r。如果只有两个数据，则 N_0 和 r 是唯一的。问题是有很多数据，而这些数据并不在同一条指数曲线上。事实上由于政策、经济、移民和战争等原因，出生率 b 和死亡率 d 并不是常数。所以用其中任何两点都不安全，我们要兼顾这些数据。于是这里使用最小二乘拟合法。由于指数函数 $\exp(t)$ 当 t 很大时可能会溢出，为了减小数值误差，首先将时间域变换至[0, 20]，所用变换为

$$t = 1800 + (t - 1800)/10$$

这样 0 代表 1800 年，1 代表 1810 年，……20 代表 2000 年，21 代表 2010 年，……r 表示 10 年增长率。另外我们需要确定 N_0 和 r 的初始值。N_0 初始值自然应取 $t=0$ 时的 N 值5.3，r 初始值取为增长率的平均值

mean(diff(N)./diff(t)./N(1:20))

为此先写 M 函数 eg8_2fun.m，其中参数 c(1)表示 N_0，c(2)表示 r，再用 M 脚本 eg8_2.m 求解。

```
%M 函数 eg8_2fun.m
    function N = eg8_2fun(c, t)
    N = c(1)* exp(c(2)* t);
```

```
%M 脚本 eg8_2.m
clear; close;
t = 0:1:20;
N = [5.3   7.2   9.6   12.9   17.1   23.2   31.4   ...
     38.6   50.2   62.9   76.0   92.0   106.5   123.2...
     131.7   150.7   179.3   204.0   226.5   251.4   275.0];
plot(t, N, 'o'); hold on;
c(1) = 5.3; c(2) = mean(diff(N)./diff(t)./N(1:20))
e0 = sum((N - eg8_2fun(c, t)).^2)
tt = [21, 22];
NN0 = eg8_2fun(c, tt)
c = curvefit('eg8_2fun', c, t, N)
e = sum((N - eg8_2fun(c, t)).^2)
NN = eg8_2fun(c, tt)
plot(tt, NN, 'r*');
tt = 0:0.1:22;
NN = eg8_2fun(c, tt);
plot(tt, NN, 'r');hold off;
```

优化结果见表 8-3。

表 8-3　人口 Malthus 模型计算结果

	初始值	拟合结果
1800 年人口 N_0(百万)	5.3	17.78
10 年增长率 r	0.2221	0.1403
残差平方和	52213	2359
2010 预测人口(百万)	562	338
2020 预测人口(百万)	702	389

图形表明中段拟合效果不错，但两头误差较大(图 8-6)。

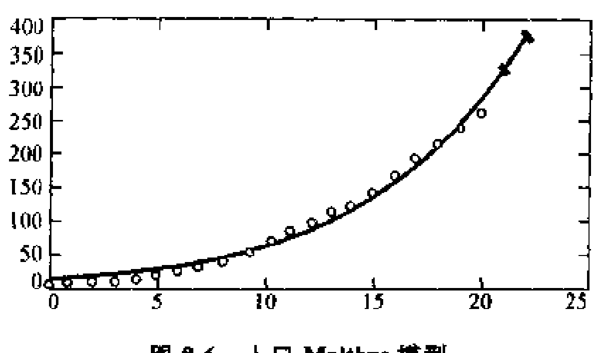


图 8-6 人口 Malthus 模型

按照 Malthus 模型，人口将呈指数增长，其缺点是没有考虑资源对人口增长的限制。Logistic 模型改进了 Malthus 模型。设 N_m 是资源容纳的最大人口数量。**Logistic 模型微分方程为**

$$N'(t) = rN(t)(1 - N(t)/N_m) \tag{8.8}$$

其中因子 $1 - N(t)/N_m$ 表示资源对人口增长阻滞因素，初值 $N(t_0) = N_0$。求解微分方程得

$$N(t) = \frac{N_m}{1 + \left(\frac{N_m}{N_0} - 1\right)e^{-r(t-t_0)}} \tag{8.9}$$

对于上述程序作相应修改，求解得表 8-4。

表 8-4 人口 Logistic 模型计算结果

	初 始 值	拟 合 结 果
1800 年人口 N_0（百万）	5.3	9.3
10 年增长率 r	0.2221	0.2910
最大人口数量 N_m	500	423
残差平方和	18380	393
2010 预测人口（百万）	266	293
2020 预测人口（百万）	293	312

可见 Logistic 模型结果较为合理（图 8-7）。

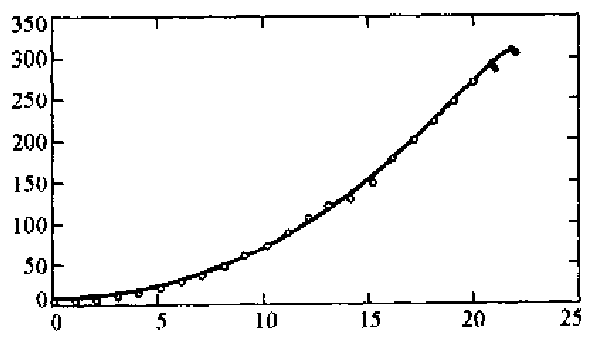


图 8-7 人口 Logistic 模型

§8.5 实验习题

1 使用分段线性插值预测例 2 中的人口，并与曲线拟合结果作比较。

2 自己编写 Lagrange 插值的 MATLAB 程序。

3 选择一些函数，在 n 个节点上(n 不要太大，可取 5～11)用 Lagrange、分段线性、三次样条三种插值方法，计算 m 个插值点的函数值(m 要适中，可取 50～100)。通过数值和图形输出，将三种插值结果与精确值进行比较。适当增加 n，再作比较，由此作初步分析。下列函数供选择参考：

(1) $y = \sin x,\ 0 \leqslant x \leqslant 2\pi$；

(2) $y = (1-x^2)^{1/2},\ -1 \leqslant x \leqslant 1$；

(3) $y = \cos^{12} x,\ -2 \leqslant x \leqslant 2$；

(4) $y = \exp(-x^2),\ -2 \leqslant x \leqslant 2$。

4 用给定的多项式，如 $y = x^3 - 6x^2 + 5x - 3$，产生一组数据 (x_i, y_i)，$i = 1, 2, \cdots, n$，再在 y_i 上添加随机干扰(可用 rand 产生(0, 1)均匀分布随机数，或用 randn 产生 $N(0, 1)$分布随机数)，然后用 x_i 和添加了随机干扰的 y_i 作 3 次多项式拟合，与原系数比较。如果作 2 或 4 次多项式拟合，结果如何?

5 假定某天的气温变化记录见表 8-5，试找出这一天的气温变化规律。

表 8-5

时刻 t(h)	0	1	2	3	4	5	6	7	8	9	10	11	12
温度℃(t)	15	14	14	14	14	15	16	18	20	22	23	25	28
时刻 t(h)	13	14	15	16	17	18	19	20	21	22	23	24	
温度℃(t)	31	32	31	29	27	25	24	22	20	18	17	16	

6 用电压 $W = 10$ 伏的电池给电容器充电，电容器上 t 时刻的电压为 $V(t) = W - (W - V_0)\exp(-t/\tau)$，其中 V_0 是电容器的初始电压，τ 是充电常数。试由表 8-6 的一组 t，V 数据确定 V_0 和 τ。

表 8-6

时间(s)	0.5	1	2	3	4	5	7	9
电压(V)	6.36	6.48	7.26	8.22	8.66	8.99	9.43	9.63

7 弹簧在力 F 的作用下伸长，一定范围内服从 Hooke 定律：F 与 x 成正比，即 $F = kx$，k 为弹性系数。现在得到表 8-7 的一组 x，F 数据，并在(x, F)坐标下作图(图 8-8)。可以看出，当 F 大到一定数值后，就不服从这个定律了。试由数据确定 k，并给出不服从 Hooke 定律时的近似公式。

表 8-7

x	1	2	4	7	9	12	13	15	17
F	1.5	3.9	6.6	11.7	15.6	18.8	19.6	20.6	21.1

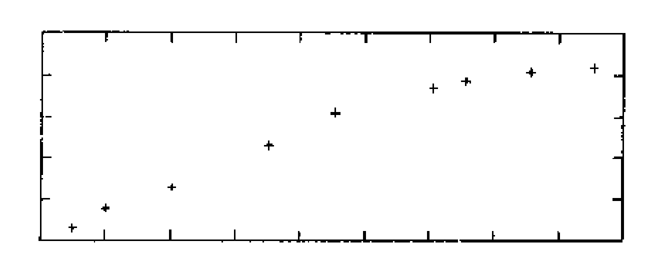


图 8-8 弹簧长度与力的关系

8 一矿脉有 13 个相邻样本点，人为地设定一原点，现测得各样本点对原点的距离 x，与样本点处某种金属含量 y 的一组数据见表 8-8，画出散点图观察二者的关系，试建立合适的回归模型，如二次曲线、双曲线、对数曲线等。

表 8-8

x	2	3	4	5	7	8	10	11	14	15	15	18	19
y	106.42	109.20	109.58	109.50	110.00	109.93	110.49	110.59	110.60	110.90	110.76	111.00	111.20

§8.6 补充知识：样条函数工具箱和二元插值

1. MATLAB 样条函数工具箱

MATLAB 样条函数工具箱(Splines)提供了丰富的样条函数工具。下面我们介绍几个插值和拟合命令。

[csape]各种边界条件的三次样条插值

> pp = csape(x, y,'边界类型', 边界值向量)，生成各种边界条件的三次样条插值。其中，边界类型可为：'complete'，给定边界一阶导数。
> 'not-a-knot'，非扭结条件，不用给边界值(默认)。
> 'periodic'，周期性边界条件，不用给边界值。
> 'second'，给定边界二阶导数。
> 'variational'，自然样条(边界二阶导数为 0)，不用给边界值。
> 可直接用 fnplt(pp)画出它的图。

我们现在用周期样条解凸轮设计问题，这样能保证整个凸轮口都是光滑的。见图 8-9，

```
»clear; close;
»x = linspace(0, 2*pi*300, 19);
»y = [502.8  525.0  514.3  451.0  326.5  188.6  92.2  59.6  62.2  102.7  ...
   147.1  191.6  236.0  280.5  324.9  369.4  413.8  458.3  502.8];
»pp = csape(x, y, 'periodic'); fnplt(pp); axis([0, 2000, 0, 550]);
»pp.breaks, pp.coefs          %分段多项式
»xi = 0:2*pi*300; yi = ppval(pp, xi)      %插值结果
```

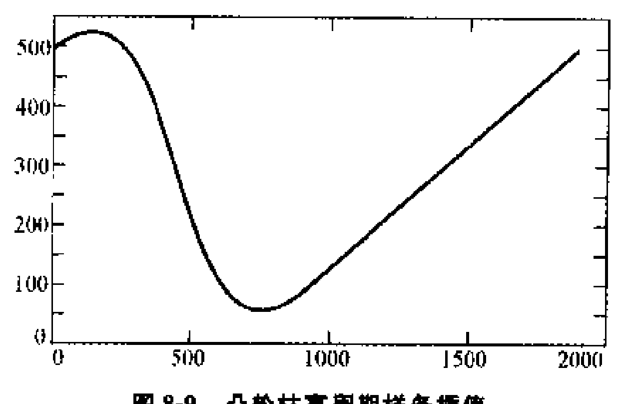


图 8-9 凸轮柱高周期样条插值

[csaps]样条拟合

pp = csaps(x, y, p) 实现光滑拟合,其中 p 为权因子,0<p<1。p 值越大,与数据越接近。特别地,若 p=0,则为线性拟合,若 p=1,则为自然样条。

当数据明显有误差,样条插值是不合适的。以下数据是带随机干扰的正弦曲线。

```
》clear; close;
》x = linspace(0, 2* pi, 21);
》y = sin(x) + (rand(1, 21) - 0.5)* 0.1;
》plot(x, y, 'o'); hold on; fnplt(csape(x, y));
```

可见,插值结果光滑性不好;

```
》fnplt(csaps(x, y, 0.8), 'r:'); hold off;
```

清除了噪声干扰。见图 8-10。

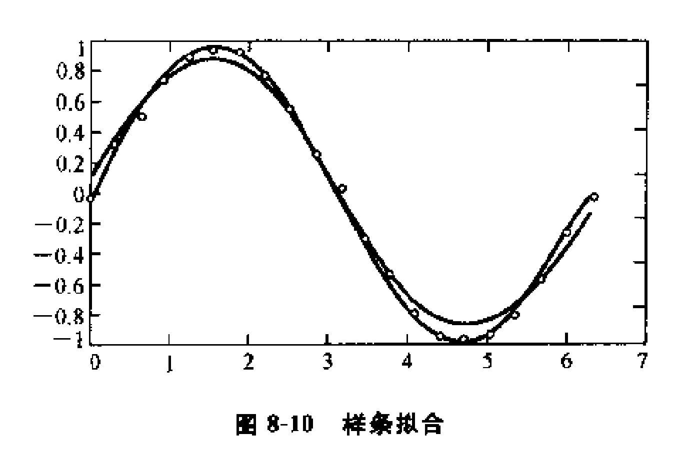


图 8-10 样条拟合

2. 多元插值

与一元函数类似,可以建立多元函数插值方法。设给定二元函数 $y = f(x, y)$ 在平面矩形格点上的函数值为

$$z_{ij} = f(x_i, y_j), i = 0, 1, \cdots, n, j = 0, 1, \cdots, m$$

二元双线性插值公式为

$$P(x, y) = \sum_{i=p}^{p+1} \sum_{j=q}^{q+1} \left(\prod_{\substack{k=p \\ k \neq i}}^{p+1} \frac{x - x_k}{x_i - x_k} \right) \left(\prod_{\substack{l=q \\ l \neq j}}^{q+1} \frac{y - y_l}{y_j - y_l} \right) z_{ij} \tag{8.10}$$

当 $x_p < x < x_{p+1}$, $y_q < y < y_{q+1}$, $p = 0, 1, \cdots, n-1$, $q = 0, 1, \cdots, m-1$

ZI = interp2(x, y, z, xi, yi)　其中，x, xi 为行向量，y, yi 为列向量，z 为矩阵。使用双线性插值。

ZI = interp2(…, 'spline')　使用二元三次样条插值。

ZI = interp2(…, 'cubic')　使用二元三次插值。

```
》clear; close; x = 0:4; y = [2:4]';
》z = [82  81  80  82  84; 79  63  61  65  81; 84  84  82  85  86];
》subplot(2, 2, 1);
》mesh(x, y, z); title('RAW DATA');
》xi = 0:0.1:4; yi = [2:0.1:4]';
》zspline = interp2(x, y, z, xi, yi, 'spline');
》subplot(2, 2, 2);
》mesh(xi, yi, zspline);
》title('SPLINE');
```

若数据是不规则的，即 z 的数据不完全，不能构成一个矩阵，从而不能直接用 interp2 插值。这时，可使用下列 griddata 命令。

ZI = griddata(x, y, z, xi, yi)　这里 x, y, z 均为向量（不必单调），表示数据，xi, yi 为网格向量。griddata 采用三角形线性插值。

ZI = griddata(x, y, z, xi, yi, 'cubic')　采用三角形三次插值。

如果上述数据残缺不全，见表 8-9。

表 8-9

y \ x	0	1	2	3	4
2	*	*	80	82	84
3	79	*	61	65	*
4	84	84	*	*	86

```
》x = [2, 3, 4, 0, 2, 3, 0, 1, 4];
》y = [2, 2, 2, 3, 3, 3, 4, 4, 4];
》z = [80, 82, 84, 79, 61, 65, 84, 84, 86];
》subplot(2, 2, 3); stem3(x, y, z); title('RAW DATA');
》ZI = griddata(x, y, z, xi, yi, 'cubic');
》subplot(2, 2, 4); mesh(xi, yi, ZI); title('GRIDDATA');
```

结果见图 8-11。

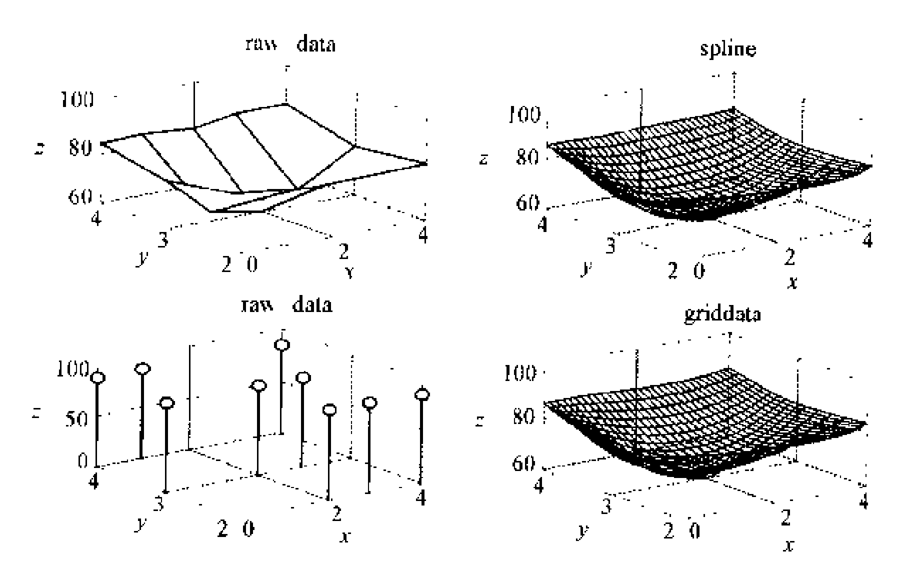


图 8-11 曲面插值

例 3 （海底测量）表 8-10 给出水面直角坐标(x, y)处水深 z，这是在低潮时测得的。如果船的吃水深度为 5 米，试问在矩形域 $75<x<200$，$-50<y<150$ 中行船应避免进入哪些区域？

表 8-10

x(m)	129	140	108	88	185	195	105
y(m)	7	111	28	147	22	137	85
z(m)	4	8	6	8	6	8	8
x(m)	157	107	77	145	162	162	117
y(m)	6	81	3	15	−66	84	−38
z(m)	9	9	8	8	9	4	9

解：我们先看看测量点的位置.

```
»clear; close;
»x=[129  140  108  88  185  195  105  157  107  77  145  162  162  117];
»y=[7  141  28  147  22  137  85  -6  -81  3  45  -66  84  -38];
»plot(x, y, 'o');
```

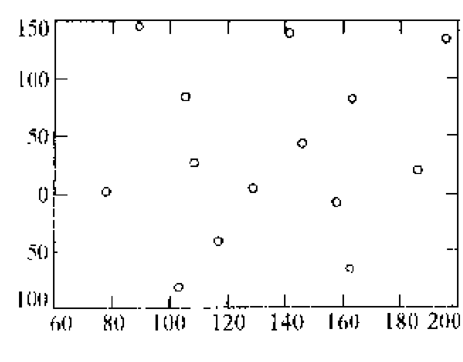


图 8-12 测量点的位置

这是一批不规则数据，见图 8-12。由于没有先验函数，我们使用插值法。为了使结果更为直观，我们考虑将 z 的数据转化为相对于海面的高度。下面绘出所考虑区域海底地形图，可以清晰看见在(129，7.5)和(162，84)附近各有一块暗礁(图 8-13)。

```
»z=[4  8  6  8  6  8  8  9  9  8  8  9  4
  9];
»h=-z;
»xi=75:5:200; yi=[-50:10:150]';
»HI=griddata(x, y, h, xi, yi, 'cubic');
```

```
»mesh(xi, yi, HI); view(-60, 30);
```

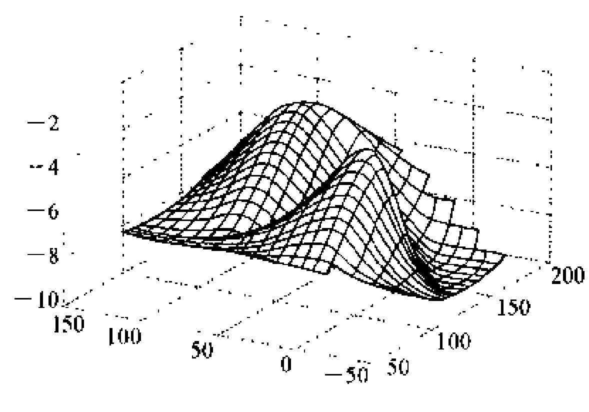


图 8-13　海底地形图

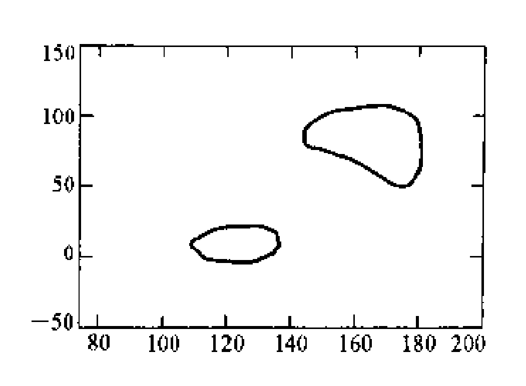


图 8-14　两个危险区域

进一步求水深不到 5 米的两个危险区域(图 8-14)。

```
»contour(xi, yi, HI, [-5, -5], 'k')
```

3. 补充习题

9　给定数据见表 8-11。

表　8-11

x	0.25	0.30	0.39	0.45	0.53
y	0.5	0.5477	0.6245	0.6708	0.7280

分别就下列端点条件求三次样条插值 $S(x)$并作图。

(i) $S'(0.25)=1$, $S'(0.53)=0.6868$;

(ii) $S''(0.25)=S''(0.53)=0$。

10　下面是一山区海拔高度每 400m 的网格数据(单位:10m)。为了作修建道路的成本预算,需要给出每 100m 的网格数据。已知山区有一山峰、一条山谷和一条溪流(其源头约 1350m),画出它们的位置。

Y \ X	0	40	80	120	160	200	240	280	320	360	400	440	480	520	560
480	135	137	139	140	141	96	94	88	80	69	57	43	29	21	1
440	137	139	141	143	144	114	111	105	95	82	69	54	38	30	21
400	138	141	143	145	147	132	128	120	108	94	78	62	46	37	35
360	142	143	145	148	150	155	151	143	130	120	98	85	75	55	50
320	143	145	116	150	155	160	155	160	160	160	155	150	150	155	155
280	95	119	137	150	120	110	155	160	155	138	107	90	105	115	120
240	91	109	127	150	120	110	135	145	120	115	101	88	100	105	110
200	88	106	123	139	150	150	140	90	110	106	95	87	90	93	95
160	83	98	118	132	145	142	140	130	70	90	85	84	38	78	75
120	74	88	108	113	125	128	123	104	90	50	70	78	75	65	55
80	65	76	88	97	102	105	102	83	80	70	30	50	55	48	35
40	51	62	73	80	85	87	85	78	72	65	50	20	30	35	32
0	37	47	55	60	67	69	67	62	58	45	40	30	10	15	25

11　在一丘陵地带测量高程,x 和 y 方向每隔 100m 测一个点,得高程见表 8 12,试拟

合一曲面，确定合适的模型，并由此找出最高点和该点的高程。

表 8-12

y \ x	100	200	300	400
100	636	697	624	478
200	698	712	630	478
300	680	674	598	412
400	662	626	552	334

12　你知道出租车的计费方式吗？收集 20～30 张当地的出租车票，以确定其计算公式。

实验九　最佳连续投资方案　线性规划

本实验的目的是通过对应用问题进行分析、建立数学模型并求解，学习掌握 MATLAB 有关线性规划求解的命令，加深对线性规划理论的理解，加强线性规划模型从“建模到求解”的数学建模主要过程的实验活动。本实验研究了连续投资、供煤量分配等应用问题。补充知识介绍了线性规划的单纯形算法，并给出了用 MATLAB 编写的单纯形算法的参考程序。

§9.1　引例：最佳连续投资方案

某部门在今后五年内考虑给下列项目投资，已知

项目 1 从第一年到第四年每年年初需要投资，并于次年末回收本利 115%；

项目 2 第三年年初需要投资，到第五年末能回收本利 125%，但规定最大投资额不超过 4 万元；

项目 3 第二年年初需要投资，到第五年末能回收本利 140%，但规定最大投资额不超过 3 万元；

项目 4 五年内每年年初可购买公债，于当年末归还，并加利息 6%。

该部门现有资金 10 万元，问它应如何确定给这些项目每年的投资额，使到第 5 年末拥有的资金的本利总额为最大？

这是一个与时间有关的连续投资问题，但在此我们对该问题不是按时间去动态地考虑，而是将五年情况总体地静态考虑。

设 y_{ij} 表示第 i 年年初投资给项目 j 的资金额度(单位：万元)，则各年的投资限制为

第一年：$y_{11} + y_{14} \leqslant 10$；

第二年：年初拥有的资金额为 $10 + 1.06y_{14} - y_{11} - y_{14}$，因此有 $y_{21} + y_{23} + y_{24} \leqslant 10 - y_{11} + 0.06y_{14}$；

第三年：年初拥有的资金额为 $10 - y_{11} + 0.06y_{14} + 1.15y_{11} + 1.06y_{24} - y_{21} - y_{23} - y_{24}$，因此有 $y_{31} + y_{32} + y_{34} \leqslant 10 + 0.15y_{11} + 0.06y_{14} - y_{21} - y_{23} + 0.06y_{24}$；

同样的分析可得：

第四年：$y_{41} + y_{44} \leqslant 10 + 0.15y_{11} + 0.06y_{14} + 0.15y_{21} - y_{23} + 0.06y_{24} - y_{31} - y_{32} + 0.06y_{34}$；

第五年：$y_{54} \leqslant 10 + 0.15y_{11} + 0.06y_{14} + 0.15y_{21} - y_{23} + 0.06y_{24} + 0.15y_{31} - y_{32} + 0.06y_{34} - y_{41} + 0.06y_{44}$。

本问题是要制定投资方案使第五年末该部门拥有的资金额最大，即

$$\max f = 1.40y_{23} + 1.25y_{32} + 1.15y_{41} + 1.06y_{54}$$

整理上述分析结果，本问题可表示为下面的数学问题

$$\max f = 1.40y_{23} + 1.25y_{32} + 1.15y_{41} + 1.06y_{54}$$

满足

$$\begin{cases} y_{11} + y_{14} \leqslant 10 \\ y_{11} - 0.06y_{11} + y_{21} + y_{23} + y_{24} \leqslant 10 \\ -0.15y_{11} - 0.06y_{14} + y_{21} + y_{23} - 0.06y_{24} + y_{31} + y_{32} + y_{34} \leqslant 10 \\ -0.15y_{11} - 0.06y_{14} - 0.15y_{21} + y_{23} - 0.06y_{24} + y_{31} + y_{32} \\ \qquad - 0.06y_{34} + y_{4} + y_{41} \leqslant 10 \\ -0.15y_{11} - 0.06y_{14} - 0.15y_{21} + y_{23} - 0.06y_{24} - 0.15y_{31} + y_{32} \\ \qquad - 0.06y_{34} + y_{41} - 0.06y_{44} + y_{54} \leqslant 10 \\ y_{ij} \geqslant 0, i = 1, 2, 3, 4, 5, j = 1, 2, 3, 4; y_{23} \leqslant 3, y_{32} \leqslant 4 \end{cases} \tag{9.1}$$

上述模型中出现的函数全为线性函数,称这样的模型为**线性规划**模型。

§9.2 线性规划基本规论复习

线性规划是运筹学中产生较早、应用广泛的一个分支。早在 20 世纪 30 年代,Канторович 研究并发表了《生产组织与计划的数学方法》,其中论述的就是线性规划问题。1947 年 Dantzing 提出了单纯形法,其后在计算机上的成功实现使得应用线性规划解决的问题迅速增加。线性规划已广泛用于国防科技、经济、工业、农业、环境工程、教育及社会科学等众多学科和领域。

1. 线性规划的提法

线性规划是研究在一组线性约束之下,某一个线性函数的最小值或最大值问题。一般地,线性规划问题的数学模型为

$$\begin{cases} \min(\text{或}\max)\ z = f_1x_1 + f_2x_2 + \cdots + f_nx_n \\ s.t. \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leqslant (\text{或} =, \text{或} \geqslant) b_1 \\ \qquad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leqslant (\text{或} =, \text{或} \geqslant) b_2 \\ \qquad \cdots \\ \qquad a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leqslant (\text{或} =, \text{或} \geqslant) b_m \\ \qquad x_i \geqslant 0 (i = 1, 2, \cdots, n) \end{cases} \tag{9.2}$$

为方便地用矩阵来简化上述记法,引入记号

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & \cdots & \cdots & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \boldsymbol{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

则线性规划问题可写为

$$
\begin{cases}
\min(\text{或}\ \max)\ z = \boldsymbol{f}'\boldsymbol{x} \\
s.t. \quad A\boldsymbol{x} \leqslant (\text{或} =, \text{或} \geqslant)\boldsymbol{b} \\
\qquad\quad x_i \geqslant 0 (i = 1,\ 2,\ \cdots,\ n)
\end{cases}
\tag{9.3}
$$

这里，$A\boldsymbol{x} \leqslant$ (或 $=$，或 $\geqslant$)$\boldsymbol{b}$ 称为**约束条件**；$z = \boldsymbol{f}'\boldsymbol{x}$ 称为**目标函数**；$\boldsymbol{x}$ 称为**决策变量**；$\boldsymbol{f}$ 称为**费用系数**；A 称为**约束矩阵**；$\boldsymbol{b}$ 为**右端向量**；$x_i \geqslant 0 (i = 1,\ 2,\ \cdots,\ n)$ 称为**非负约束**。其中 $s.t.$ 是 subject to 的缩写，意思是"满足约束条件"。

2. 线性规划问题的标准形式

线性规划问题的标准形式为

$$
\begin{cases}
\min(\text{或}\ \max)\ z = \boldsymbol{f}'\boldsymbol{x} \\
s.t. \quad A\boldsymbol{x} = \boldsymbol{b} \\
\qquad\quad x_i \geqslant 0 (i = 1,\ 2,\ \cdots,\ n)
\end{cases}
\tag{9.4}
$$

其中 $\boldsymbol{x}$ 为**决策向量**，$\boldsymbol{f}$ 为**常数向量**，$\boldsymbol{b}$ 为**非负常数向量**，A 为**常数矩阵**。

任何一种线性规划模型都可以等价地转换为标准形式。

(1) 约束条件标准化——**松弛变量法**

如果约束条件中有不等式

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leqslant b_i$$

$$a_{j1}x_1 + a_{j2}x_2 + \cdots + a_{jn}x_n \geqslant b_j$$

可通过引人两个非负变量 x_{n+1}、x_{n+2} 将上述条件写成下面等价形式

$$
\begin{cases}
a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + x_{n+1} = b_i \\
x_{n+1} \geqslant 0
\end{cases}
$$

$$
\begin{cases}
a_{j1}x_1 + a_{j2}x_2 + \cdots + a_{jn}x_n - x_{n+2} = b_j \\
x_{n+2} \geqslant 0
\end{cases}
$$

可见约束不等式可转换为约束等式。

(2) 自由变量的标准化

线性规划标准形式中要求决策变量非负，对于不满足该条件的变量 x_i，可通过引进非负变量 x_i'，x_i''，并增加约束条件 $x_i = x_i' - x_i''$ 或在表达式中直接以 $x_i' - x_i''$ 替代 x_i 即可。

(3) 目标函数的标准化

若原问题是求 $\max z = \boldsymbol{f}'\boldsymbol{x}$，可以转换为求 $\min(-z) = -\boldsymbol{f}'\boldsymbol{x}$。

3. 关于线性规划的概念与名词

在(9.4)式中满足约束条件的向量 $\boldsymbol{x} = (x_1,\ x_2,\ \cdots,\ x_n)'$ 称为线性规划的**可行解**，所有可行解的集合称为**可行域**，使目标函数 $z = \boldsymbol{f}'\boldsymbol{x}$ 达到最小的可行解称为**最优解**。

如果矩阵 A 的某 m 列所构成的方阵 B 是满秩的，则称 B 为线性规划问题的一个**基**，A 的剩余部分组成的子矩阵记为 N，即 A 可以写成 $A = (B,\ N)$。$\boldsymbol{x}$ 相应地写成 $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_B \\ \boldsymbol{x}_N \end{pmatrix}$，$\boldsymbol{x}_B$ 的分量与 B 的列相对应，称为**基变量**。$\boldsymbol{x}_N$ 的变量与 N 的列相对应，称为**非基变量**。在约束

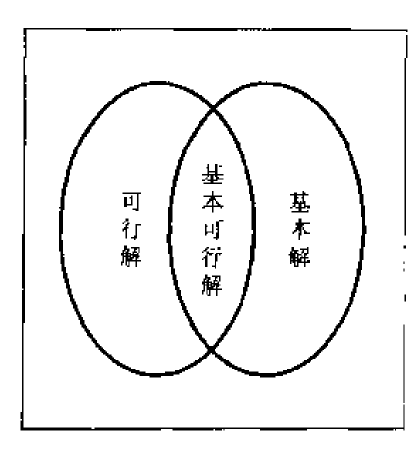


图 9-1 线性规划解集合

$Ax=b$ 中令所有的非基变量取值为零时，得到的解 $x=\begin{pmatrix}x_B\\x_N\end{pmatrix}=\begin{pmatrix}B^{-1}b\\0\end{pmatrix}$ 称为相应于 B 的基本解。

基本解的基变量都取非负值时，即满足 $x_B\geqslant 0$，则称该基本解为**基本可行解**。相应的基 B 称为**可行基**。既是最优解又是其可行解的 x 称为基**最优解**。

上述几个概念可以用图 9-1 来表示其关系。

(1) 若线性规划问题有最优解，则必至少在某个基可行解上找到。

(2) 若两个基可行解 x^*，x^{**} 都是最优解，则它们的凸组合 $x=\lambda x^*+(1-\lambda)x^{**}$ $(0\leqslant\lambda\leqslant 1)$ 也是**最优解**，此即最优解有无穷多的情况。

§9.3 求解线性解划的 MATLAB 命令

1. MATLAB5.2 以下版本命令 lp

求解线性规划问题

$$\begin{cases}\min f'x\\ s.t.\ \ Ax\leqslant b\end{cases}\tag{9.5}$$

这里 A 为 $m\times n$ 矩阵，f 为 $m\times 1$ 向量，b 为 $m\times 1$ 向量。

```
x = lp(f, A, b)  求解线性规划(9.5);
x = lp(f, A, b, vlb, vub)  指定上下界 vlb≤x≤vub;
x = lp(f, A, b, vlb, vub, x0)  给定迭代初值 x0;
x = lp(f, A, b, vlb, vub, x0, n)  前 n 个约束为等式。
    可以用 help lp 查阅有关该命令的详细信息。
```

2. MATLAB5.3 解本(本实验使用的版本)命令 linprog

该版本推出的优化工具箱(optimization Toolbox 2.0)较老版本作了相当大的修改，虽然仍保留了命令 lp，但已使用新的命令 linprog 取代 lp。

```
x = linprog(f, A, b)  求解线性规划(9.5);
x = linprog(f, A, b, Aeq, beq)  求解下面线性规划
                    min z=f'·x, A·x≤b, Aeq·x=beq;
x = linprog(f, A, b, Aeq, beq, lb, ub)  指定 lb≤x≤ub;
x = linprog(f, A, b, Aeq, beq, lb, ub, x0)
```

指定迭代初值 x0,如果没有不等式约束,可用[]替代 A 和 b 表示缺省;如果没有等式约束,可用[]替代 Aeq 和 beq 表示缺省;如果某个 x_i 下无界或上无界,可设定 lb(i) =-inf 或 ub(i) = inf;
用[x, Fval]代替上述各命令行中左边的 x,则可得到最优解处的函数值 Fval。
可以用 help linprog 查阅有关该命令的详细信息。

例 1　解下列线性规划

$$\begin{cases} \min z = -5x_1 - 4x_2 - 6x_3 \\ s.t. \quad x_1 - x_2 + x_3 \leqslant 20 \\ \qquad 3x_1 + 2x_2 - 4x_3 \leqslant 42 \\ \qquad 3x_1 + 2x_2 \leqslant 30 \\ \qquad x_i \geqslant 0, \ i = 1, 2, 3 \end{cases}$$

```
》f=[-5; -4; -6]; A=[1, -1, 1; 3, 2, 4; 3, 2, 0];
》b=[20; 42; 30]; lb=zeros(3, 1);
》[x, feval]=linprog(f, A, b, [], [], lb)
Optimization terminated successfully.
x=
    0.0000
   15.0000
    3.0000
feval=
  -78.0000
```

§9.4 实验例题

例 2 (最佳连续投资方案)为便于输入计算,在引例最佳连续投资方案的数学模型中,根据实际使用的变量进行下列对比转换

y_{11}	y_{14}	y_{21}	y_{23}	y_{24}	y_{31}	y_{32}	y_{34}	y_{41}	y_{44}	y_{54}
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}

解:计算如下

```
》clear
》f=zeros(1, 11);
》f(9)=1.15; f(4)=1.4; f(7)=1.25; f(11)=1.06;
》A=[1, 1, zeros(1, 9);1, -0.06, 1, 1, 1, zeros(1, 6);
```

```
    -0.15, -0.06, 1, 1, -0.06, 1, 1, 1, zeros(1, 3);
    -0.15, -0.06, -0.15, 1, -0.06, 1, 1, -0.06, 1, 1, 0;
    -0.15, -0.06, -0.15, 1, -0.06, -0.15, 1, -0.06, 1,...
    -0.06, 1];
》b = 10* ones(5, 1);
》lb = zeros(11, 1); ub = inf* ones(11, 1); ub([4, 7]) = [3; 4];
》[x, fval] = linprog( - f, A, b, [], [], lb, ub)        %以上是输入
    %下面是输出
Optimization terminated successfully.
x=
    5.8406
    4.1594
    1.4090
    3.0000
    0.0000
    1.3098
    4.0000
    1.4069
    3.1117
    0.0000
    1.5062
fval=
  -14.3750
```

由此可见最佳连续投资方案为：

第一年：$y_{11} = 58406$ 元，$y_{14} = 41594$ 元；

第二年：$y_{21} = 14090$ 元，$y_{23} = 30000$ 元，$y_{24} = 0$ 元；

第三年：$y_{31} = 13098$ 元，$y_{32} = 40000$ 元，$y_{34} = 14069$ 元；

第四年：$y_{41} = 31117$ 元，$y_{44} = 0$ 元；

第五年：$y_{54} = 15062$ 元；

第五年末该部门拥有资金总额为 143750 元，盈利 43.75%。

例 3 （供煤量分配）某两个煤厂 A_1 和 A_2 每月进煤量分别为 60t(吨)和 100t，联合供应 3 个居民区 B_1、B_2 和 B_3。3 个居民区每月对煤的需求量依次分别为 50t、70t、40t。煤厂 A_1 离 3 个居民区 B_1、B_2 和 B_3 的距离分别为 10km、5km 和 6km，煤厂 A_2 离 3 个居民区 B_1、B_2 和 B_3 的距离分别为 4km、8km 和 12km。问如何分配供煤量使得运输量(即 t×km)达到最小？

解：以总运输量为目标函数，记为 f，煤厂供给居民的煤量为决策变量，记为 x_{ij}（表示煤厂 A_i($i = 1, 2$) 提供给居民区 B_j($j = 1, 2, 3$) 的煤量），则根据"A_i 运出煤 = A_i 运进煤"和"B_j 从 A_i 运进煤量之和 = B_j 需求量"可得下述线性规划问题

$$
\begin{cases}
\min f = 10x_{11} + 5x_{12} + 6x_{13} + 4x_{21} + 8x_{22} + 12x_{23} \\
s.t. \quad x_{11} + x_{12} + x_{13} = 60 \\
\qquad x_{21} + x_{22} + x_{23} = 100 \\
\qquad x_{11} + x_{21} = 50 \\
\qquad x_{12} + x_{22} = 70 \\
\qquad x_{13} + x_{23} = 40 \\
\qquad x_{ij} \geqslant 0, i = 1, 2; j = 1, 2, 3
\end{cases}
$$

求解如下

```
》c = [10; 5; 6; 4; 8; 12];
》Aeq = [1, 1, 1, 0, 0, 0; 0, 0, 0, 1, 1, 1; 1, 0, 0, 1, 0, 0;
     0, 1, 0, 0, 1, 0; 0, 0, 1, 0, 0, 1];
》beq = [60; 100; 50; 70; 40];
》[x, fval] = linprog(c, [], [], Aeq, beq, zeros(6, 1))
Optimization terminated successfully.   %输出
x=
    0.0000
   20.0000
   40.0000
   50.0000
   50.0000
    0.0000
fval=
  940.0000
```

这里的 940 为最小运输量。

§9.5 实 验 习 题

1 求解下面的线性规划问题

$$
\begin{cases}
\min f = -3x_1 + 4x_2 - 2x_3 + 5x_4 \\
s.t. \quad 4x_1 - x_2 + 2x_3 - x_4 = -2 \\
\qquad x_1 + x_2 + 3x_3 - x_4 \leqslant 14 \\
\qquad -2x_1 + 3x_2 - x_3 + 2x_4 \geqslant 2 \\
\qquad x_1, x_2, x_3 \geqslant 0, x_4 \text{ 无约束}
\end{cases}
$$

2 求解线性规划问题

$$
\begin{cases}
\min f = 5x_1 + 4x_2 + 8x_3 \\
s.t. \quad x_1 + 2x_2 + x_3 = 6 \\
\quad -2x_1 + x_2 \geqslant -4 \\
\quad 5x_1 + 3x_2 \leqslant 15 \\
\quad x_j \geqslant 0, j = 1, 2, 3
\end{cases}
$$

3 （生产计划制定）某工厂制造甲、乙两种产品，每种产品消耗煤、电、工作日及获利如表 9-1 所示，现有煤 360t(吨)，电力 200kW·h，工作日 300 个。请制定一个使总利润最大的生产计划。

表 9-1

	煤(t)	电(kW·h)	工 作 日	单位利润(元/t)
甲	9	4	3	7000
乙	5	5	10	12000

4 （制定配棉方案）棉纺厂的主要原料是棉花，一般要占总成本的 70%左右。所谓配棉问题，就是要根据绵纱的质量指标，采用各种价格不同的棉花，按一定的比例配制成纱，使其既达到质量指标，又使总成本最低。

棉纱的质量指标一般由棉结和品质指标来决定。这两项指标都可用数量形式来表示。一般来说，棉结粒数越少越好，品质指标越大越好。

一个年纺纱能力为 15000 锭的小厂在采用最优化方法配棉前，某一种产品 32D 纯棉纱的棉花配比、质量指标及单价见表 9-2。

有关部门对 32D 纯棉纱规定的质量指标为棉结不多于 70 粒，品质指标不小于 2900。请给出配棉方案。

表 9-2

原 料 品 名	单价(元/t)	混合比(%)	棉结(粒)	品质指标	混棉单价(元/t)
国棉 131	8400	25	60	3800	2100
国棉 229	7500	35	65	3500	2625
国棉 327	6700	40	80	2500	2680
平均合计		100	70	3175	7405

提示：可考虑使混棉的单价最小。

§9.6 补充知识：单纯形算法

1. 单纯形法原理

对于标准形式的线性规划问题

$$
\begin{cases}
\min(\text{或} \max) \ f = \boldsymbol{c}'\boldsymbol{x} \\
s.t. \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\
\quad x_i \geqslant 0 (i = 1, 2, \cdots, n)
\end{cases}
$$

若有有限最优值，则目标函数的最优值必在某一基本可行解处达到，因而只要在基本可行解中寻找最优解。这就使我们有可能用穷举法来求得线性规划问题的最优解，但当变量很多时计算量很大，有时行不通。**单纯形法**(simplex method)的基本思想就是先找一个基本可行解，检验是否为最优解或判断问题无解。否则，再转换到另一个使目标函数值减小的基可行解上，重复上述过程，直至求到问题的最优解或指出问题无解。

原理如下：设找到初始基本可行解$\bar{x}$，可行基为 B，非基矩阵为 N，即可写 $A=(B,\ N)$。于是$\bar{x}=\begin{pmatrix}B^{-1}\boldsymbol{b}\\ \boldsymbol{0}\end{pmatrix}=\begin{bmatrix}\bar{\boldsymbol{b}}\\ \boldsymbol{0}\end{bmatrix}$。相应地，目标函数值为 $\bar{f}=\boldsymbol{c}\bar{x}=(\boldsymbol{c}_B,\ \boldsymbol{c}_N)\begin{bmatrix}\bar{\boldsymbol{b}}\\ \boldsymbol{0}\end{bmatrix}=\boldsymbol{c}_B\bar{\boldsymbol{b}}$，其中 $\boldsymbol{c}_B$ 是 $\boldsymbol{c}$ 中与基变量 $\boldsymbol{x}_B$ 对应的分量组成的 m 维行向量。

再设任意可行解 $\boldsymbol{x}=\begin{bmatrix}\boldsymbol{x}_B\\ \boldsymbol{x}_N\end{bmatrix}$，由 $A\boldsymbol{x}=\boldsymbol{b}$ 得到

$$\boldsymbol{x}_B=B^{-1}\boldsymbol{b}-B^{-1}N\boldsymbol{x}_N=\bar{\boldsymbol{b}}-B^{-1}N\boldsymbol{x}_N \tag{9.6}$$

相应的目标函数值为

$$f=\boldsymbol{c}\boldsymbol{x}=\boldsymbol{c}_B\bar{\boldsymbol{b}}-(\boldsymbol{c}_BB^{-1}N-\boldsymbol{c}_N)\boldsymbol{x}_N \tag{9.7}$$

若记 $A=(\boldsymbol{a}_1,\ \boldsymbol{a}_2,\ \cdots,\ \boldsymbol{a}_n)$，则有

$$f=\bar{f}-\sum_{j\in N_B}(\boldsymbol{c}_BB^{-1}\boldsymbol{a}_j-c_j)x_j \tag{9.8}$$

其中，N_B 是非基变量的下标集。记

$$z_j-c_j=\boldsymbol{c}_BB^{-1}\boldsymbol{a}_j-c_j \tag{9.9}$$

称为检验数，于是有

$$f=\bar{f}-\sum_{j\in N_B}(z_j-c_j)x_j \tag{9.10}$$

变换后的问题叙述如下

$$\begin{cases}\min\ f=\bar{f}-\sum\limits_{j\in N_B}(z_j-c_j)x_j\\ s.t.\quad \boldsymbol{x}_B+B^{-1}N\boldsymbol{x}_N=\bar{\boldsymbol{b}}\\ \qquad\quad x_j\geqslant 0,\ j=1,\ \cdots,\ n\end{cases} \tag{9.11}$$

其中，$\bar{f}$是基本可行解$\bar{x}$所对应的目标函数值。

若基本可行解的所有基变量都取正值，则称它是**非退化的**；若有取零值的基变量，则称它是退化的。称所有基本可行解都非退化的线性规划为非退化的。

对于非退化的线性规划(9.11)，有下面的结论

(a) 若所有 $z_j-c_j\leqslant 0$，则 $\bar{x}$为问题(9.4)的最优解，记为 $\boldsymbol{x}^*$。

(b) 若 $z_k-c_k>0$，$k\in N_B$，且相应的 $B^{-1}\boldsymbol{a}_k\leqslant 0$，则问题(9.4)无有界最优解。

(c) 若 $z_k-c_k>0$，$k\in N_B$，且$\bar{\boldsymbol{a}}_k=B^{-1}\boldsymbol{a}_k$ 至少有一个正分量，则能找到基本可行解 $\hat{\boldsymbol{x}}$，使目标数值下降，即 $\boldsymbol{c}\hat{\boldsymbol{x}}<\boldsymbol{c}\bar{\boldsymbol{x}}$。

由上述三个结论可以得到单纯形算法的基本思路：首先找一个基本可行解$\bar{x}$并求出与之相应的检验数z_j-c_j，若所有的$z_j-c_j\leqslant 0$，则$\bar{x}$为最优解；若存在k使$z_k-c_k>0$，且相应的$B^{-1}a_k\leqslant 0$，则问题无有界最优解；若存在k使$z_k-c_k>0$，且相应的$B^{-1}a_k$含有正的分量，则可以求出一个新的基本可行解$\hat{x}$，使得目标函数值减少。然后，再对$\hat{x}$重复以上做法，如此经过有限次迭代后一定能找到最优解或判定问题无有界最优解。

为求出一个新的基本可行解$\hat{x}$，可取正的检验数中最大的z_k-c_k的下标k所对应的A的列向量a_k进入基，使相应的非基变量x_k取正值变为基变量。可令

$$x_k=\min\left\{\frac{\bar{b}_i}{\bar{a}_{ik}}\middle|\bar{a}_{ik}>0\right\}\triangleq\frac{\bar{b}_r}{\bar{a}_{rk}}$$

并且，让原来的基变量x_{B_r}变为非基变量，即令$x_{B_r}=0$。这样得到新的基本可行解为

$$\hat{x}=(\hat{x}_{B_1},\cdots,\hat{x}_{B_{r-1}},0,\hat{x}_{B_{r+1}},\cdots,\hat{x}_{B_m},0,\cdots,x_k,\cdots,0)'$$

相应的目标函数值为

$$\hat{f}=c\hat{x}=\bar{f}-(z_k-c_k)x_k<\bar{f}$$

且有

$$\hat{x}_{B_i}=\bar{b}_i-\bar{a}_{ik}x_k\geqslant 0\ (i=1,\cdots,r-1,r+1,\cdots,m)$$

$$\hat{x}_{B_r}=x_k>0$$

即$\hat{x}_B\geqslant\mathbf{0}$。

由上述原理可有如下算法。

2. 单纯形算法步骤

对标准形式的线性规划问题，单纯形的计算步骤如下

(i) 找初始可行基B和初始基本可行解。

(ii) 求出$\boldsymbol{x}_B=B^{-1}\boldsymbol{b}\triangleq\bar{\boldsymbol{b}}$，计算目标函数值$f=\boldsymbol{c}_B\boldsymbol{x}_B$。

(iii) 计算检验数z_j-c_j，$j=1,2,\cdots,n$，并按$z_k-c_k=\max\{z_j-c_j\mid j=1,2,\cdots,n\}$确定下标$k$，取$x_k$为进基变量。

(iv) 若$z_k-c_k\leqslant 0$，停止，此时基本可行解$\boldsymbol{x}=\begin{pmatrix}\boldsymbol{x}_B\\ \boldsymbol{x}_N\end{pmatrix}=\begin{pmatrix}\bar{\boldsymbol{b}}\\ \mathbf{0}\end{pmatrix}$是最优解，

目标函数最大值为$f=\boldsymbol{c}_B\bar{\boldsymbol{b}}$；否则，执行下步。

(v) 计算$\bar{\boldsymbol{a}}_k=B^{-1}\boldsymbol{a}_k$，若$\bar{\boldsymbol{a}}_k\leqslant\mathbf{0}$，停止。此时问题无有界解；否则执行下步。

(vi) 求最小比$\dfrac{\bar{b}_r}{\bar{a}_{rk}}=\min\left\{\dfrac{\bar{b}_i}{\bar{a}_{ik}}\middle|\bar{a}_{ik}>0\right\}$(*)确定下标$r$，取$x_{B_r}$为离基变量。

(vii) 以$\boldsymbol{a}_k$代替$\boldsymbol{a}_{B_r}$得到新基，并令$x_k=\dfrac{\bar{b}_r}{\bar{a}_{rk}}$再返回执行(ii)。

按上述步骤用MATLAB编写线性规划单纯形算法程序lps. m

%M 函数 lps. m

```
function [x, f] = lps(c, A, b)
tb = find(b<0); b(tb) = - b(tb); A(tb,:) = - A(tb, :);
[m, n] = size(A); B = A(:, 1:m); x = zeros(n, 1); m1 = 1:m;
```

```
while det(B) = = 0|~isempty(find(inv(B)* b<0))
    temp = randperm(n); m1 = temp(1:m); B = A(:, m1);
end
xB = B\b; x(m1) = xB; f - c(m1)* x(m1);
criterior = c(m1)* (B\A) - c; [z1, z2] = max(criterion);
while z1>0
    az2 = B\A(:, z2);
    if az2< = 0, disp('问题无界'),break
else, t1 = find(az2>0);
      [tt1, tt2] = min(xB(t1)./az2(t1));
      t3 = t1(tt2); B(:, t3) = A(:, z2); x(m1) = xB - tt1* az2;
      m1(t3) = z2; x(z2) = tt1; f = c(m1)* xB; xB = x(m1);
      criterion = (c(m1)/B)* A - c;
      [z1, z2] = max(criterion);
    end; end
x(m1) = xB; f = c* x;
```

程序中用的是随机搜索方法寻找初始可行基 B。

例 4　用单纯形法求解下列线性规划

$$\begin{cases}\min\ x_1 + x_2 - 4x_3 \\ s.t.\ \ x_1 + x_2 + 2x_3 + x_4 = 9 \\ \qquad x_1 + x_2 - x_3 + x_5 = 2 \\ \qquad -x_1 + x_2 + x_3 + x_6 = 4 \\ \qquad x_j \geqslant 0(j = 1,\ 2,\ \cdots,\ 6)\end{cases}$$

解：

```
》c = [1, 1, -4, 0, 0, 0]; b = [9; 2; 4];
》a = [1, 1, 2, 1, 0, 0; 1, 1, -1, 0, 1, 0; -1, 1, 1, 0, 0, 1];
》x = lps(c, a, b)
x =
    0.3333
         0
    4.3333
         0
    6.0000
         0。
```

如果在基本可行解中存在有基变量为零的情况，则称为**退化的基本可行解**。在基本可行解退化时，有可能发生用单纯形法要进行**无限多次迭代**也得不到最优解的**死循环**。

例如用上述程序计算下面线性规划问题时则出现了死循环。

例 5

$$\begin{cases}\min -0.75x_4+20x_5-0.5x_6+6x_7\\ s.t.\ \ x_1+0.25x_4-8x_5-x_6+9x_7=0\\ \quad x_2+0.5x_4-12x_5-0.5x_6+3x_7=0\\ \quad x_3+x_6=1\\ \quad x_j\geqslant 0(j=1,2,\cdots,7)\end{cases}$$

而 $\boldsymbol{x}=(0.75,0,0,1,0,1,0)'$ 是该问题的解。

避免出现这种循环的一种方法是字典序规则法

(i) 若向量 $\boldsymbol{x}\neq\boldsymbol{0}$，且它的第一个非零分量是正的，则称 $\boldsymbol{x}$ 是按字典序正的，记为 $\boldsymbol{x}\succ\boldsymbol{0}$；若 $\boldsymbol{x}=\boldsymbol{0}$ 或是按字典序正的，则称 $\boldsymbol{x}$ 是按字典序非负的，记为 $\boldsymbol{x}\succeq\boldsymbol{0}$。

(ii) 对向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$，有 $\boldsymbol{x}-\boldsymbol{y}\succ\boldsymbol{0}$，则称 $\boldsymbol{x}$ 是按字典序大于 $\boldsymbol{y}$，记为 $\boldsymbol{x}\succ\boldsymbol{y}$；若有 $\boldsymbol{x}-\boldsymbol{y}\succeq\boldsymbol{0}$，则记为 $\boldsymbol{x}\succeq\boldsymbol{y}$。

(iii) 若在一组 $\boldsymbol{x}^{(i)}$ 中，存在 k 使得 $\boldsymbol{x}^{(i)}\succeq\boldsymbol{x}^{(k)}$，则称 $\boldsymbol{x}^{(k)}$ 为这组向量中按字典序最小的，记成 $\boldsymbol{x}^{(k)}=\text{lex}\min\limits_{i}\boldsymbol{x}^{(i)}$。

编制求字典序最小向量的 MATLAB 程序 lex _ min.m

```
            %M 函数 lex _ min.m
function [y, k] = lex _ min(x)
% [y, k] = lex _ min(x)按行求矩阵 x 字典序最小行向量
% 返回值 y 是矩阵 x 字典序最小行向量,k 是 y 在 x 中的行数
[mx, nx] = size(x); k = 1; y = x(1, :);
if mx = = 1, k = 1; y = x(1, :);
else,
    [t1, t2] = min(x); t3 = zeros(mx, nx);
    for I = 1:nx; t3(:, I) = t1(I)* ones(mx, 1); end
    t4 = sum((t3~ = x)); tt = find(t4~ = 0);
    k = t2(tt(1)); y = x(k, :);
end
```

例 6　用上述程序判定 $\boldsymbol{x}=(0,0,3,-1,2)$，$\boldsymbol{y}=(0,2,4,0,3)$ 按字典序的正负并比较 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 按字典序的大小如下

```
》x = [0, 0, 3, -1, 2]; y = [0, 2, 4, 0, 3]; z = zeros(1, 5);
》a = lex _ min([x; z]); b = lex _ min([y; z]);
》c = lex _ min([x; y]); Comparison = [a; b; c]
Comparison=
     0     0     0     0     0
     0     0     0     0     0
     0     0     3    -1     2
```

第一、二行表示 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 都是按字典序正的，$\boldsymbol{x}$ 按字典序小于 $\boldsymbol{y}$。

按字典序规则，单纯形法迭代不会出现上述用 lps.m 计算例 5 的循环。做法是在单纯形算法中将步骤 vi 中的(＊)式换成下式即可。记 $pp=\bar{b}B^{-1}$，p_i 表示 pp 的第 i

个行向量：$\frac{\boldsymbol{p}_r}{\bar{a}_{rk}} = \text{lex}\min\left\{\frac{\boldsymbol{p}_i}{\bar{a}_{ik}} \middle| \bar{a}_{ik} > 0\right\}$。

3. 采用大 M 法及字典序的单纯形算法参考程序(用 MATLAB 编写)

用单纯形法解线性规划问题时，需先有一个初始可行基和解。为解决这个问题可采用随机搜索方法寻找初始可行基 B，但有时会使算法不稳定(运用上面的程序 lps.m 计算有时会出现不稳定)。可采用如下的大 M 法来解决这个问题。

在约束中引入人工变量 $\boldsymbol{x}_a = (x_{n+1}, x_{n+2}, \cdots, x_{n+m})^T$，并在目标函数中加上惩罚项 $M\boldsymbol{e}\boldsymbol{x}_a$(其中 $\boldsymbol{e} = (1, 1, \cdots, 1)$)，变原线性规划问题为

$$\begin{cases} \min(\text{或 }\max)\ f = \boldsymbol{c}\boldsymbol{x} + M\boldsymbol{e}\boldsymbol{x}_a \\ s.t.\ \ A\boldsymbol{x} + \boldsymbol{x}_a = \boldsymbol{b} \\ \qquad x_i \geqslant 0 (i = 1, 2, \cdots, n+m) \end{cases}$$

其中 M 是足够大的正数。以下是采用大 M 法和字典序规则用 MATLAB 编写的单纯形算法的参考程序 lps _ Mlex.m

```
                 %M 函数 lps _ Mlex.m
function [x, f] = lps _ Mlex(c, A, b, M, N, pre)
%   [x, f] = lps _ Mlex(c, A, b, M, N, pre)采用单纯形算法中的
%   大 M 法，并结合字典排序规则解下列线性规划
%   min f = c'* x   s.t. Ax = b, x 所有分量≥0
%   M 是一充分大的数，N 是引进人工变量的个数，N 应
%   不超过(通常等于)约束等式的个数，pre 是精度。
%   返回结果 x 是最优解，f 是最优解处的函数值。
[m, n] = size(A);
if nargin<6, pre = 0; end; if nargin<5, N = 0; end
if N>m, error('N 不能超过约束条件的个数 m');
    else,
    A = [A, [zeros(N, m - N); eye(m - N)]];
    c = [c(:)', zeros(1, m - N)];
    A = [A, eye(m)]; c = [c, M* ones(1, m)];
    m1 = n + m - N + 1:n + 2* m - N; B = A(:, m1);
    x = zeros(n + 2* m - N, 1); x = x(:);
    tb = find(b<0); b(tb) = - b(tb); A(tb, :) = - A(tb, :);
    xB = B\b; x(m1) = xB; f = c* x;
    criterion = c(m1)* (B\A) - c; [z1, z2] = max(criterion);
while z1>pre
    az2 = B\A(:, z2);
    if az2< = pre, x = nan* ones(length(c)); break
    else, t1 = find(az2>pre); p = [xB, B\eye(size(B))]; pp = [];
        for kk = 1:length(t1);
```

```
            pp(kk, :) = p(t1(kk), :)./az2(t1(kk));
        end
        tt1 = min(xB(t1)./az2(t1)); [tt0, tt2] = lex_min(pp);
        t3 = t1(tt2); B(:, t3) = A(:, z2); x(m1) = xB - tt1*az2;
        m1(t3) = z2; x(z2) = tt1; f = c(m1)*xB; xB = x(n1);
        criterion = c(m1)*(B\A) - c; [z1, z2] = max(criterion);
    end, end, end;
if sum(x(n + m - N + 1:n + 2*m - N)) <= pre*m
    x(m1) = xB; f = c(1:n)*x(1:n); x = x(1:n);
else, x = nan*ones(length(c)); x = x(:); x = x(1:n);
end
```

下面的例子我们调用 lps_Mlex.m 来求解。

例 7　用 lps_Mlex.m 来求解前面的例 5，看是否出现死循环。

解：

```
》clear
》f = [0; 0; 0; -3/4; 20; -0.5; 6];
》a = [0.25, -8, -1, 9; 0.5, -12, -0.5, 3; 0, 0, 1, 0];
》a = [eye(3), a]; b = [0; 0; 1]; vlb = zeros(7, 1);
》x = linprog(f, [], [], a, b, vlb);
》xx = lps_Mlex(f, a, b, 100000, 3); [x, xx]
Optimization terminated successfully.
ans =          %输出结果
    0.7500      0.7500
    0.0000           0
    0.0000           0
    1.0000      1.0000
    0.0000           0
    1.0000      1.0000
    0.0000     -0.0000
```

可见用 lps_Mlex 与 linprog 结果一样，没有出现死循环。

例 8　（合金工厂的生产计划）某合金厂生产甲、乙两种合金，生产每 1t(吨)甲和乙种合金各需用 A、B、C 三种元素的量见表 9-3。

表　9-3

	需 A 元素(kg)	需 B 元素(kg)	需 C 元素(kg)
甲(t^{-1})	20	40	90
乙(t^{-1})	100	80	60

工厂每月所能获得的 A、B 和 C 三种元素最大供应量分别为 200kg、200kg 和 360kg。

工厂生产甲种合金的利润为 30 万元/t，生产乙种合金的利润为 40 万元/t。工厂该如何制定生产计划，才能获得最大利润？

解：设工厂每月生产甲和乙种合金分别为 x_1(t) 和 x_2(t)，所获利润为 f 万元，则目标函数为：$f=30x_1+40x_2$。由题意容易得到下列线性规划问题

$$\begin{cases}\max\ f=30x_1+40x_2\\ s.t.\ \ 20x_1+100x_2\leqslant 200\\ \qquad 40x_1+80x_2\leqslant 200\\ \qquad 90x_1+60x_2\leqslant 360\\ \qquad x_1,\ x_2\geqslant 0\end{cases}$$

标准化为下列模型

$$\begin{cases}\min\ -f=-30x_1-40x_2-0\cdot x_3-0\cdot x_4-0\cdot x_5\\ s.t.\ \ 20x_1+100x_2+x_3=200\\ \qquad 40x_1+80x_2+x_4=200\\ \qquad 90x_1+60x_2+x_5=360\\ \qquad x_1,\ x_2\geqslant 0\end{cases}$$

调用 lps _ Mlex.m 求解如下

```
》clear;
》A=[20, 100, 1, 0, 0; 40, 80, 0, 1, 0; 90, 60, 0, 0, 1];
》b=[200; 200; 360]; c= -[30; 40; 0; 0; 0];
》[x, f]=lps _ Mlex(c', A, b, 10000)
x=          %输出结果
     3.5000
     0.7500
    55.0000
          0
          0
f=
   -135
```

可见工厂应生产甲和乙两种合金分别为 3.5t 和 0.75t，最大利润为 135 万元。

4. 补充习题

5 用单纯形法(或调用上面程序 lps _ Mlex.m)求解线性规划

$$\begin{cases}\min\ z=-x_1-2x_2+x_3-x_4-4x_5+2x_6\\ s.t.\ \ x_1+x_2+x_3+x_4+x_5+x_6\leqslant 6\\ \qquad 2x_1+x_2-2x_3+x_4\leqslant 4\\ \qquad x_3-x_4+2x_5+x_6\leqslant 4\\ \qquad x_j\geqslant 0,\ j=1,2,\cdots,6\end{cases}$$

实验十　飞行管理问题　非线性规划

本实验的目的是通过对应用问题分析、建立数学模型并求解，学习掌握 MATLAB 有关非线性规划求解的命令，加深对非线性规划理论的理解，加强数学建模解决实际问题关键过程的实验活动。本实验研究了飞行管理和公司营业计划等应用问题。补充知识介绍了非线性规划的一些基本算法。

§10.1　引例：飞行管理问题

这是一个 1995 年全国大学生数学建模竞赛题，问题如下：

在约 10000m 高空的某边长为 160km 的正方形区域内，经常有若干架飞机作水平飞行。区域内每架飞机的位置、方向角和速度由计算机记录数据，以便进行飞行管理。当一架欲进入该区域的飞机到达区域边缘时，记录其数据后，要立即计算并判断是否会与区域内的飞机发生碰撞。如果会碰撞，则应计算如何调整各架（包括新进入的）飞机飞行的方向角，以避免碰撞。现假定条件如下

(1) 不碰撞的标准为任意两架飞机的距离大于 8km；

(2) 飞机飞行方向角调整的幅度不应超过 30°；

(3) 所有飞机飞行速度均为 800km/h；

(4) 进入该区域的飞机在到达区域边缘时，与区域内飞机的距离应在 60km 以上；

(5) 最多需考虑 6 架飞机；

(6) 不必考虑飞机离开此区域的状况。

请你对避免碰撞的飞行管理问题建立数学模型，列出计算步骤，对以下数据进行计算（方向角误差不超过 0.01°）。要求飞机飞行方向角调整的幅度尽量小。

设该区域 4 个顶点的坐标为

(0, 0), (160, 0), (160, 160), (0, 160)。

记录数据见表 10-1。

表 10-1　飞行管理问题

飞机编号	横坐标	纵坐标	方向角度
1	150	140	243°
2	85	85	236°
3	150	155	220.5°
4	145	50	159°
5	130	150	230°
新进入	0	0	52°

注：方向角指飞行方向与 x 轴正向的夹角。

此问题很容易想到以各飞机调整的飞行角度平方和作为目标函数，而以每两架飞机之间的最小距离不超过 8km、各飞机飞行角度调整的值不超过 30°为约束条件。如此得出的是一个非线性模型，在计算上可能会复杂些。

以 t 表示时间，x_i 与 y_i 分别表示第 i 架飞机的横纵坐标（问题中已经给出），θ_i 表示第 i 架飞机的飞行方向角（问题中已经给出），$d_{ij}(t)$ 表示 t 时刻第 i 架飞机与第 j 架飞机间的距离，v 表示飞机的飞行速度（$v=800$）。

则目标函数为：

$$f = \sum_{i=1}^{6} \Delta\theta_i^2$$

$$d_{ij}^2(t) = (x_i - x_j + v\,t(\cos(\theta_i + \Delta\theta_i) - \cos(\theta_j + \Delta\theta_j)))^2$$
$$+ (y_i - y_j + v\,t(\sin(\theta_i + \Delta\theta_i) - \sin(\theta_j + \Delta\theta_j)))^2$$

则约束条件为

$$D_{ij} \triangleq \min_{t \geqslant 0} d_{ij}^2(t) > 64,\ i,\ j = 1,\ \cdots,\ 6,\ i \neq j$$

$$\frac{\mathrm{d}d_{ij}^2}{\mathrm{d}t} = 0 \Rightarrow t = -a/b$$

其中

$$a = (x_i - x_j)(\cos(\theta_i + \Delta\theta_i) - \cos(\theta_j + \Delta\theta_j))$$
$$+ (y_i - y_j)(\sin(\theta_i + \Delta\theta_i) - \sin(\theta_j + \Delta\theta_j))$$

$$b = v[(\cos(\theta_i + \Delta\theta_i) - \cos(\theta_j + \Delta\theta_j))^2$$
$$+ (\sin(\theta_i + \Delta\theta_i) - \sin(\theta_j + \Delta\theta_j))^2]$$

将 t 代入即可求出 D_{ij}。于是，本问题的一个数学模型为

$$\begin{cases} \min\ f = \sum_{i=1}^{6} \Delta\theta_i^2 \\ s.t.\ \ D_{ij} > 64 \\ \qquad |\Delta\theta_i| \leqslant \frac{\pi}{6} \\ \qquad i,\ j = 1,\ \cdots,\ 6,\ i \neq j \end{cases}$$

引入记号

$\boldsymbol{\Delta\theta} = (\Delta\theta_1,\ \cdots,\ \Delta\theta_6)^T$，$\boldsymbol{g} = (g_1,\ \cdots,\ g_{15})^T$（$\boldsymbol{g}$ 是由 $64 - D_{ij}$ 按 $i,\ j = 1,\ \cdots,\ 6,\ i \neq j$ 构成的向量，在下面的程序中计算），则模型为

$$\begin{cases} \min\ f = \boldsymbol{\theta}'\boldsymbol{\theta} \\ s.t.\ \ \boldsymbol{g} < \boldsymbol{0} \\ \quad \mathbf{vlb} \leqslant \boldsymbol{\theta} \leqslant \mathbf{vub} \end{cases} \tag{10.1}$$

其中 $\mathbf{vlb} = -\frac{\pi}{6}(1,\ 1,\ 1,\ 1,\ 1,\ 1)^T$，$\mathbf{vub} = \frac{\pi}{6}(1,\ 1,\ 1,\ 1,\ 1,\ 1)^T$，$T$ 为转置。

像这样在模型的目标函数或约束条件中出现了非线性函数，称为**非线性规划**（本模型是**约束非线性规划**）。

§10.2 非线性规划基本理论复习

很多实际问题所归结的优化数学模型中，目标函数或约束条件很难用线性函数表达。如果目标函数或约束条件中包含有非线性函数，就称这种优化模型为**非线性规划**问题，解这种问题要用非线性规划方法。由于很多实际问题要求进一步精确化，以及计算机的发展，使非线性规划在近 30 多年来得以迅速发展，并在最优化设计、管理科学系统得到越来越广泛的应用。

非线性规划问题要比线性规划问题难得多，非线性规划有着众多算法，而且仍有不少算法不断提出，但它却不像线性规划有单纯形法这一通用方法。

1. 问题提法

非线性规划处理的问题是在等式或不等式约束下优化某个目标函数，求出最优解。一般可表示成

$$\begin{cases} \min f(\boldsymbol{x}) \\ s.t.\ g_i \geqslant 0,\ i = 1, 2, \cdots, m \\ \quad h_j(\boldsymbol{x}) = 0,\ j = 1, 2, \cdots, k \end{cases} \tag{10.2}$$

其中 $x \in \mathrm{R}^n$，$f(\boldsymbol{x})$ 为**目标函数**，$g_i(\boldsymbol{x})$，$h_j(\boldsymbol{x})$ 为**约束函数**，这些函数中，至少有一个是非线性函数。

$$令\ S = \{\boldsymbol{x} \mid g_i(\boldsymbol{x}) \geqslant 0,\ i = 1, 2, \cdots, m;\ h_j(\boldsymbol{x}) = 0,\ j = 1, 2, \cdots, k\}$$

称 S 为**可行集**或**可行域**，S 中的点称为**可行点**。这样(10.2)式中的约束条件可用集约束表示

$$\min f(\boldsymbol{x}),\ s.t.\ \boldsymbol{x} \in S \tag{10.3}$$

特别称

$$\min f(\boldsymbol{x}),\ \boldsymbol{x} \in \mathrm{R}^n \tag{10.4}$$

此为**无约束优化**问题。

2. 相关的基本知识

下面给出最优解及一些相关的概念。

(1) 设 $f(\boldsymbol{x})$ 为目标函数，S 为可行域，$\boldsymbol{x}^* \in S$，若对每一个 $x \in S$ 均成立 $f(\boldsymbol{x}) \geqslant f(\boldsymbol{x}^*)$，则称 $\boldsymbol{x}^*$ 为极小化问题(10.3)式的最优解(**整体最优解**)；若存在 $\boldsymbol{x}^*$ 的某邻域，使得对该邻域中的每个 x 成立 $f(\boldsymbol{x}) \geqslant f(\boldsymbol{x}^*)$，则称 $\boldsymbol{x}^*$ 为极小化问题的**局部最优**解。对于极大化问题，可类似地定义**整体**最优解和**局部**最优解。

(2) 称向量 $\nabla f(\boldsymbol{x}) = \left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \frac{\partial f(\boldsymbol{x})}{\partial x_2}, \cdots, \frac{\partial f(\boldsymbol{x})}{\partial x_n}\right)^T$ 为 $f(\boldsymbol{x})$（如果一阶偏导数存在）在点 $\boldsymbol{x}$ 处的**梯度**；称矩阵

$$\nabla^2 = \begin{pmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n^2} \end{pmatrix}$$

为 $f(\boldsymbol{x})$（如果二阶偏导数存在）在点 $\boldsymbol{x}$ 处的 **Hesse 矩阵**。

(3) 无约束问题的极值条件，考虑无约束问题(10.4)。

局部极小点的一阶必要条件：设函数 $f(\boldsymbol{x})$ 在点 $\bar{\boldsymbol{x}}$ 处可微，若 $\bar{\boldsymbol{x}}$ 是局部极小点，则梯度 $\nabla f(\bar{\boldsymbol{x}}) = \boldsymbol{0}$。

局部极小点的二阶必要条件：设函数 $f(\boldsymbol{x})$ 在点 $\bar{x}$ 处二次可微，若 $\bar{x}$ 是局部极小点，则梯度 $\nabla f(\bar{x}) = \boldsymbol{0}$，且 Hesse 矩阵半正定。

局部极小点的充分条件：设函数 $f(x)$ 在点 $\bar{x}$ 处二次可微，若梯度 $\nabla f(\bar{x}) = \boldsymbol{0}$，且 Hesse 矩阵正定，则 $\bar{x}$ 是局部极小点。

(4) 约束问题的最优性条件，考虑约束问题(10.2)，S 是可行域。

下面介绍几个经常用到的概念。

(i) 设向量 $\boldsymbol{d}$ 是 $\bar{x}$ 点的任一方向，若存在正数 δ，使得对每个实数 $\lambda \in (0, \delta)$，都有 $f(\bar{x}+\lambda \boldsymbol{d}) < f(\bar{x})$，则称 $\boldsymbol{d}$ 为函数 $f(x)$ 在 $\bar{x}$ 处的**下降方向**。如果 $f(x)$ 是可微函数，且 $\nabla f(\bar{x})^T \boldsymbol{d} < \boldsymbol{0}$，则 $\boldsymbol{d}$ 必为 $f(x)$ 在 $\bar{x}$ 处的下降方向。

(ii) 设 $\bar{x}$ 是非线性规划问题(10.2)的一个可行点，向量 $\boldsymbol{d}$ 是 $\bar{x}$ 点的某一方向，若存在正数 δ，使得对每个实数 $\lambda \in (0, \delta)$，都有 $\bar{x} + \lambda \boldsymbol{d} \in S$，则称 $\boldsymbol{d}$ 为 $\bar{x}$ 点的一个**可行方向**。

(iii) 对问题(10.2)一个可行解 $\bar{x}$，对于约束 $g_i(\bar{x}) \geq 0$，如等式成立，则称该约束是在 $\bar{x}$ 点的一个**起作用约束**。通常将所有等式成立的约束条件一并称为在 $\bar{x}$ 处的起作用约束。起作用约束在 $\bar{x}$ 的邻域限制了可行点的范围，也就是说，当点沿某些方向稍微离开 $\bar{x}$ 时，仍能满足这些约束条件；而沿着另一些方向离开 $\bar{x}$ 时，不论步长多么小，都将违背这些约束条件；所有在 $\bar{x}$ 处严格不等式成立的约束条件一并称为在 $\bar{x}$ 处的**不起作用约束**。

局部极小点的一阶必要条件

(i) **Fritz John 条件**：设在问题(10.2)中，$\bar{x}$ 为可行点，$I = \{i \mid g_i(\bar{x}) = 0\}$，$f$ 和 $g_i (i \in I)$ 在 $\bar{x}$ 可微，$g_i (i \notin I)$ 在 $\bar{x}$ 点连续，$h_j (j = 1, 2, \cdots, k)$ 在 $\bar{x}$ 连续可微。如果 $\bar{x}$ 是局部最优解，则存在不全为零的数 w_0，w_i 和 $v_j (i \in I, j = 1, 2, \cdots, k)$，使得

$$w_0 \nabla f(\bar{x}) - \sum_{i \in I} w_i \nabla g_i(\bar{x}) - \sum_{j=1}^{k} v_j \nabla h_j(\bar{x}) = \boldsymbol{0}, \quad w_0, w_i \geq 0, i \in I$$

通常将满足 Fritz John 条件的点称为 **Fritz John 点**。

(ii) **Kuhn-Tucker 条件**：设 $\bar{x}$ 为问题(10.2)的可行点，$I = \{i \mid g_i(\bar{x}) = 0\}$，$f$ 和 $g_i (i \in I)$ 在 $\bar{x}$ 可微，$g_i (i \notin I)$ 在 $\bar{x}$ 点连续，$h_j (j = 1, \cdots, k)$ 在 $\bar{x}$ 连续可微，向量集

$$\{\nabla g_i(\bar{x}), \nabla h_j(\bar{x}) \mid i \in I, j = 1, 2, \cdots, k\}$$

线性无关，如果 $\bar{x}$ 是局部最优解，则存在非负数 w_i 和数 $v_j (i \in I, j = 1, 2, \cdots, k)$，使得

$$\nabla f(\bar{x}) - \sum_{i \in I} w_i \nabla g_i(\bar{x}) - \sum_{j=1}^{k} v_j \nabla h_j(\bar{x}) = \mathbf{0}$$

由上述条件知，在最优解处，目标的梯度可用起作用约束梯度的非负线性组合及等式约束梯度的线性组合来表示，上述条件还可写成等价形式：$\nabla f(\bar{x}) - \sum_{i=1}^{m} w_i \nabla g_i(\bar{x}) - \sum_{j=1}^{k} v_j \nabla h_j(\bar{x}) = \mathbf{0}$，$w_i g_i(\bar{x}) = 0$，$i = 1, \cdots, m$。后一等式称为**互补松弛条件**。

(iii) Lagrange 函数表述的一阶条件：称

$L(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{v}) = f(\boldsymbol{x}) - \sum_{i=1}^{m} w_i \nabla g_i(\boldsymbol{x}) - \sum_{j=1}^{k} v_j \nabla h_j(\boldsymbol{x})$ 为 Lagrange 函数，其中 $\boldsymbol{w} = (w_1, \cdots, w_m)^T$ 和 $\boldsymbol{v} = (v_1, \cdots, v_k)^T$ 称为 **Kuhn-Tucker 乘子**，也称为 **Lagrange 乘子**。一阶必要条件可以表达为

$$\begin{cases} \nabla_x L(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{v}) = 0 \\ g_i(\boldsymbol{x}) \geqslant 0, \ i = 1, \cdots, m \\ h_j(\boldsymbol{x}) = 0, \ j = 1, \cdots, k \\ w_i g_i(\boldsymbol{x}) = 0, \ i = 1, \cdots, m \\ w_i \geqslant 0, \ i = 1, \cdots, m \end{cases} \tag{10.5}$$

(5) 凸集与凸函数

(i) 凸集：R^n 中集合 S，若满足：对任意两点 $\boldsymbol{x}^{(1)}$ 和 $\boldsymbol{x}^{(2)}$ 及每个实数 $\lambda \in [0, 1]$，都有 $\lambda \boldsymbol{x}^{(1)} + (1-\lambda)\boldsymbol{x}^{(2)} \in S$，则称 S 为**凸集**，$\lambda \boldsymbol{x}^{(1)} + (1-\lambda)\boldsymbol{x}^{(2)}$ 称为 $\boldsymbol{x}^{(1)}$ 和 $\boldsymbol{x}^{(2)}$ 的**凸组合**。

(ii) 凸函数：设 S 为 R^n 中的非空凸集，f 是 S 上的实函数，如果对任意的 $\boldsymbol{x}^{(1)}$，$\boldsymbol{x}^{(2)}$ 及每个数 $\lambda \in (0, 1)$ 都有

$f(\lambda \boldsymbol{x}^{(1)} + (1-\lambda)\boldsymbol{x}^{(2)}) \leqslant \lambda f(\boldsymbol{x}^{(1)}) + (1-\lambda) f(\boldsymbol{x}^{(2)})$，则称 f 为 S 上的**凸函数**。如果对任意相异的两点，上面严格不等式成立，则称 f 为 S 上的严格凸函数；如果 $-f$ 为 S 上的凸函数，则称 f 为 S 上的**凹函数**。

(iii) 凸函数的一阶判别条件：R^n 中非空开凸集 S 上的可微函数 f 是凸函数的充要条件为对任意两点 $\boldsymbol{x}^{(1)}$，$\boldsymbol{x}^{(2)} \in S$，成立

$f(\boldsymbol{x}^{(2)}) \geqslant f(\boldsymbol{x}^{(1)}) + \nabla f(\boldsymbol{x}^{(1)})^T(\boldsymbol{x}^{(2)} - \boldsymbol{x}^{(1)})$。

(iv) 凸函数的二阶判别条件：R^n 中非空开凸集 S 上的二次可微函数 f 是凸函数的充要条件为在每一点 $\boldsymbol{x} \in S$ 处 Hesse 阵是半正定的。

(v) 凸规划：若在(10.2)式中 f 是凸函数，g_i 是凹函数，h_j 是线性函数，则问题(10.2)为求凸函数在凸集上的极小点，这类问题称为**凸规划**。凸规划是非线性规划中的一种重要特殊情形，它具有很好的性质，正如定理所述，凸规划的局部极小点就是整体极小点。如果凸规划的目标函数是严格凸函数，又存在极小点，则它的极小点是唯一的。

§ 10.3 求解非解性解划的 MATLAB 命令

1. MATLAB 5.2 及以下版本(在 MATLAB5.3 也可使用)

(1) 无约束极值

fmin、fmins 参见实验三。

也可用 help 命令来了解上述命令更多的用法。

(2) 约束求极值

constr:求解下列形式非线性规划问题

$$\min f=f(x),\ s.t.\ g(x)\leqslant 0,\ vlb\leqslant x\leqslant vub$$

```
x = constr('fun',初值,选项, vlb, vub),
其中初值、选项、vlb 和 vub 可以省略,或用[]替代以表示
省略或取默认值,'fun 写成如下的 M 函数 fun.m 形式
                         function  [f, g] = fun(x)
                            f = f(x); g = g(x);
可调用 help constr 来了解 constr 更多的用法。
```

2. MATLAB 5.3 版本

fminbnd:采用黄金分割和抛物线插值法,求解一元函数极值。

fminsearch:采用 Nelder-Meade 单纯形搜索法,求解多元函数极值。

fminunc:采用 Newton 法求解多元函数极值。

用法如下

```
x = fminbnd(函数名,x1, x2,选项),
    其中'函数名'为目标函数,用 M 函数文件表示。
    求解范围为 x1<x<x2,选项可省略。
[x, f] = fminbnd(...)同时返回解 x 处的函数值。
x = fminsearch(函数名,初值,选项),
    其中'函数名'为目标函数,用 M 函数文件表示。选项可省略。
[x, f] = fminsearch(...)同时返回解 x 处的函数值。
fminunc:用法和 fminsearch 一样。
可用 help 命令来了解上述命令更多的用法。
```

fmincon:求解下列形式非线性规划问题

$$\min f(x)\ s.t.\ \begin{cases} A\cdot x\leqslant b,\ Aeq\cdot x=beq, \\ c(x)\leqslant 0,\ ceq(x)=0,\ lb\leqslant x\leqslant ub \end{cases}$$

根据约束条件,调用格式分别为

```
x = fmincon(fun,初值, A, b);
x = fmincon(fun,初值,A, b, Aeq, beq);
x = fmincon(fun,初值,A, b, Aeq, beq, lb, ub);
x = fmincon(fun,初值,A, b, Aeq, beq, lb, ub, nonlcon);
```

当约束条件中缺 A 和 b、Aeq 和 beq 或 lb 和 ub 时，
相关项可用[]代替。fun 写成如下的 M 函数形式(fun.m)

```
function  f = fun(x)
f = f(x);
```

非线性约束条件写成如下的 M 函数形式(nonlcon.m)

```
function [c, ceq] = nonlcon(x)
c = c(x); ceq = ceq(x);
```

[x, f] = fmincon(...)同时返回解 x 处的函数值。
可调用 help fmincon 来了解 fmincon 更多的用法。

注：若原问题是求 $\max f$,可以转换为求 $\min(-f)$。

§10.4 实验例题

例 1 (飞行管理问题)按非线性规划模型(10.1),调用 MATLAB 命令 fmincon 求解，先写两个 M 函数 airfun.m 和 airfunco.m 如下

```
                                %M 函数 airfun.m
function f = airfun(delta)
f = delta*delta';
                                %M 函数 airfunco.m
function [c, ceq] = airfunco(delta)
% 约束函数 airfunco.m
x0 = [150  85  150  145  130  0]; y0 = [140, 85, 155, 50, 150, 0];
alpha0 = [243  236  220.5  159  230  52]*Pi/180; v = 800;
co = cos(alpha0 + delta);
si = sin(alpha0 + delta);
for i = 2:6
    for j = 1:i - 1
        t(i, j) = (x0(i) - x0(j))*(co(i) - co(j));
        t(i, j) = t(i, j) + (y0(i) - y0(j))*(si(i) - si(j));
        t(i, j) = -t(i, j)/v;
        t(i, j) = t(i, j)/((co(i) - co(j))^2 + (si(i) - ...
si(j))^2);
        if t(i, j)<0, d(i, j) = 1000;
        else,
            d(i, j) = (x0(i) - x0(j) + v*t(i, j)*(co(i) - ...
co(j)))^2;
            d(i, j) = (y0(i) - y0(j) + v*t(i, j)*(si(i) - ...
si(j)))^2 + d(i, j);
```

```
end; end; end;
c = 64 - [d(2, 1), d(3, 1:2), d(4, 1:3), d(5, 1:4), d(6, 1:5)];
ceq = [];
```

由于非线性规划求解对初值依赖性较大，我们可在零点随机生成若干个初值来获取可能的最优解。为此可编写一个 M 文件 eg10 _ 1. m 如下。

%M 脚本 eg10 _ 1. m

```
clear;
deltaini = [0, 0, 0, 0, 0, 0];
  vlb = - Pi/6* ones(1, 6); vub = Pi/6* ones(1, 6);
options = optimset('LargeScale', 'off');
[d1, fval] = fmincon('airfun', - deltaini, [], [], [], [], ...
vlb, vub, 'airfunco', options);
n = 10
for i = 1:n - 1
    deltaini = [(rand(1, 6) - 0.5)* 0.1];
    [dt, feval] = fmincon('airfun', deltaini, [], [], [], [], ...
vlb, vub, 'airfunco', options);
    if feval<fval, fval = feval; d1 = dt; end;
end;
d1, fval
```

在 MATLAB 命令窗口计算如下

```
》eg10 _ 1
n=
      10      %十个初值
d1=           %最优解
  0.0000   0.0000   0.0360   -0.0086   -0.0001   0.0273%弧度
fval=         %最优解函数值
  0.0021
```

注：上述程序在奔腾 133 内存为 16 兆的 PC 机上运行时间大约 10 多秒，而如调用 constr 在零初值点大约用不到 3 秒的时间即可求出上述结果。在此对计算时间就不做特殊限制和讨论。有关此问题的其他解法可参见《数学的实践与认识》1996(1)。

例 2　求函数 $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ 的最小值点，初始点为 [-1.2, 1]。

解：调用命令 fminsearch 求解

```
》[x, fval] = fminsearch('100* (x(2) - x(1)^2)^2 + (1 - x(1))^2', [-1.2, 1])   %输入
Optimization terminated successfully: %输出
the current x satisfies the termination criteria using OPTIONS. TolX of
1.000000e-004
and F(X) satisfies the convergence criteria using OPTIONS. TolFun of
```

1.000000e-004

x=

 1.0000 1.0000

fval=

 8.1777e-010

也可调用命令 fminunc 求解

»[x, fval] = fminunc('100*(x(2)-x(1)^2)^2+(1-x(1))^2', [-1.2, 1]) %输入

Optimization terminated successfully: %输出

Current search direction is a descent direction, and magnitude of directional

derivative in search direction less than 2* options. TolFun

x=

 1.0000 1.0000

fval=

 1.9118e-011

例 3 求函数 $f = -1.5x_1^2 - 0.5x_2^2 + x_1x_2 + 2x_1$ 极大值点。

解: %M 函数:ffun.m

```
function y = ffun(x)
y = 1.5*x(1)^2 + 0.5*x(2)^2 - x(1)*x(2) - 2*x(1); % 转换求 max 为求 min
```

求解如下

```
»x1 = fminsearch('ffun', [-2, 4]);
»x2 = fminunc('ffun', [-2, 4]); x1, x2 % 输入
```

Optimization terminated successfully: %输出

the current x satisfies the termination criteria using OPTIONS. TolX of

1.000000e-004

and F(X) satisfies the convergence criteria using OPTIONS. TolFun of

1.000000e-004

x1=

 1.0000 1.0000

Optimization terminated successfully:

Search direction less than 2* options. TolX

x2=

 1.0000 1.0000

例 4 某公司经营两种物品,第一种物品每吨(t)售价 30 元,第二种物品售价 450 元/t,根据统计,售出每吨第一种物品所需要的营业时间平均是 0.5h(小时),第二种物品是 $2+0.25x_2$(h),其中 x_2 是第二种物品售出的数量。已知该公司在这段时间内的总营业时间为 800h,试决定使其营业额最大的营业计划。

解:设该公司经营第一种设备 x_1 件,第二种设备 x_2 件,则营业额为

$f = 30x_1 + 450x_2$,则

$$\begin{cases}\max f = 30x_1 + 450x_2 \\ s.t.\ 0.5x_1 + 2x_2 + 0.25x_2^2 \leqslant 800 \\ \quad x_1 \geqslant 0,\ x_2 \geqslant 0\end{cases}$$

先写两个 M 函数

%M 函数 fffun. m

```
function f = fffun(x)
f = -30*x(1) - 450*x(2);
```

%M 函数 fcon. m

```
function[c, ceq] = fcon(x)
c = 0.5*x(1) + 2*x(2) + 0.25*x(2)^2 - 800; ceq = [];
```

求解如下

```
》[x, fval] = fmincon('fffun', [0; 0], [], [], [], [], [0; 0], [inf; inf],'fcon')
Optimization terminated successfully: Magnitude of directional derivative in search direction less than 2*options. TolFun and maximum constraint violation is less than options.TolCon
Active Constraints:
     3
x=
  1.0e+003*
     1.4955
     0.0110
fval=
-4.9815e+004
```

问题得解，最大营业额为 $-f = 49815$。

§10.5 实验习题

1 求解 $\min(x_1^4 + (x_2+1)^2)$，初始点为 $x = (1,1)'$。

2 求解 $\max(-2x_1^2 - 2x_1x_2 - 5x_2^2)$，取初始点 $x = (2, -2)'$。

3 求解 $\min((6 + x_1 + x_2)^2 + (2 - 3x_1 - 3x_2 - x_1x_2)^2)$，初始点 $x = (-4, 6)'$。

4 设有 400 万元资金，要求 4 年内用完，若在一年内使用资金 x 万元，则可得到利润 $\sqrt{x}$万元(利润不能再使用)，当年不用的资金可存入银行，年利率为 10%。试制订出资金的使用规划，以使 4 年利润为最大。

5 求 $\max f(x) = x_1x_2x_3$，$s.t. \begin{cases}-x_1 + 2x_2 + 2x_3 \geqslant 0 \\ x_1 + 2x_2 + 2x_3 \leqslant 72 \\ 10 < x_2 < 20 \\ x_1 - x_2 = 10\end{cases}$

6 某工厂向用户提供一种产品，按合同规定，其交货数量和日期是：第一季度末交 40t

(吨)，第二季度末交60t，第三季度末交80t。工厂的最大生产能力为每季度100t，每季度的生产费用是$f(x)=50x+0.2x^2$(元)，此处x为该季度生产该产品的吨数。若工厂生产的多，多余的该产品可移到下季度向用户交货，这样，工厂就需支付存储费，每吨该产品每季度的存储费为4元。问该厂每季度应生产多少吨该产品，才能既满足交货合同，又使工厂所花费的费用最少(假定第一季度开始时该产品无存货)。

§10.6 补充知识：非线性规划算法

1. 一维搜索

求解非线性规划所用的计算方法，最常见的是迭代下降法算法。其一般步骤是，得到点$\boldsymbol{x}^{(k)}$后，按某种规则确定一个方向$\boldsymbol{d}^{(k)}$，从$\boldsymbol{x}^{(k)}$出发沿此方向在直线(或射线)上求目标函数的极小点，从而得到$\boldsymbol{x}^{(k)}$的后继点$\boldsymbol{x}^{(k+1)}$，再从$\boldsymbol{x}^{(k+1)}$出发重复以上步骤，直至求出问题的解，这种方法称为**一维搜索**，或**线搜索**。

一维搜索可归结为单变量函数的极小化问题。设目标函数为$f(\boldsymbol{x})$，过点$\boldsymbol{x}^{(k)}$沿方向$\boldsymbol{d}^{(k)}$的直线可用点集来表示，记作$L=\{\boldsymbol{x}\mid \boldsymbol{x}=\boldsymbol{x}^{(k)}+\lambda\boldsymbol{d}^{(k)},\ -\infty<\lambda<\infty\}$。求$f(\boldsymbol{x})$在直线$L$上的极小点转化为求一元函数$\varphi(\lambda)=f(\boldsymbol{x}^{(k)}+\lambda\boldsymbol{d}^{(k)})$的极小点。为此可采用**黄金分割法**，具体算法可参见前面的实验。

2. 无约束非线性规划

考虑问题(10.4)求无约束问题的极小点，一般通过一系列一维搜索来实现。其核心问题是选择搜索方向，搜索方向不同则形成不同的最优化方法。无约束问题的最优化方法一般分作两类：一类在计算过程中使用导数，可称为**使用导数的最优化方法**，亦称为**解析法**；另一类在计算过程中只用到目标函数，通常称为**直接方法**。**最速下降法**(亦称**梯度法**)是解析法中最为古老的一种方法，同时也是理解某些其他最优化方法的基础。

最速下降法(梯度法)

最速下降法由法国数学家Cauchy于1847年首先提出。这种方法，在每次迭代中，沿最速下降方向(负梯度方向)进行搜索。迭代公式为

$$\begin{cases}\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\lambda_k\boldsymbol{d}^{(k)}\\ \boldsymbol{d}^{(k)}=-\nabla f(\boldsymbol{x}^{(k)})\\ \lambda_k:f(\boldsymbol{x}^{(k)}+\lambda_k\boldsymbol{d}^{(k)})=\min\limits_{\lambda\geqslant 0}f(\boldsymbol{x}^{(k)}+\lambda\boldsymbol{d}^{(k)})\end{cases}$$

算法如下

(i) 给定初点$\boldsymbol{x}^{(1)}$，允许误差$\varepsilon>0$，置$k=1$。

(ii) 求搜索方向$\boldsymbol{d}^{(k)}=-\nabla f(\boldsymbol{x}^{(k)})$。

(iii) 若$\|\boldsymbol{d}^{(k)}\|<\varepsilon$，则停止计算；否则，从$\boldsymbol{x}^{(k)}$出发沿$\boldsymbol{d}^{(k)}$进行一维搜索，求$\lambda_k$使得

$$f(\boldsymbol{x}^{(k)}+\lambda_k\boldsymbol{d}^{(k)})=\min_{\lambda\geqslant 0}f(\boldsymbol{x}^{(k)}+\lambda\boldsymbol{d}^{(k)})。$$

(iv) 令$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\lambda_k\boldsymbol{d}^{(k)}$，置$k:=k+1$，转(ii)。

最速下降法在一定条件下是收敛的。最速下降法产生的序列是线性收敛的，而且收敛性质与极小点处Hesse矩阵$\nabla^2 f(\bar{x})$的特征值有关。

设$f(\boldsymbol{x})$存在连续二阶偏导数，$\bar{x}$是局部极小点，Hesse矩阵$\nabla^2 f(\bar{x})$的最小特征值$a>$

0,最大特征值为 A,若令 $r=\dfrac{A}{a}$,则条件数越小,算法产生的序列收敛越快,条件数越大,收敛性越慢。

3. 约束非线性规划

(1) 罚函数法

考虑约束问题(10.2),其中

$$f(x),\ g_i(x)(i=1,\cdots,m),\ h_j(x)(j=1,\cdots,k)$$

是连续函数。罚函数法的基本思想是,利用目标函数和约束函数组成辅助函数 $F(\boldsymbol{x},\sigma)=f(\boldsymbol{x})+\sigma P(\boldsymbol{x})$。$F(\boldsymbol{x},\sigma)$ 具有这样的性质:当点 $\boldsymbol{x}$ 位于可行域以外时,函数 $F(\boldsymbol{x},\sigma)$ 取值很大,而且离可行域越远其值越大;当点 $\boldsymbol{x}$ 在可行域内时,函数 $F(\boldsymbol{x},\sigma)=f(\boldsymbol{x})$。这样,可将原来问题转化成关于辅助函数 $F(\boldsymbol{x},\sigma)$ 的无约束极小值问题

$$\min F(\boldsymbol{x},\sigma)\triangleq f(\boldsymbol{x})+\sigma P(\boldsymbol{x}) \tag{10.6}$$

在极小化过程中,若 $\boldsymbol{x}$ 不是可行点,则辅助函数中的第2项 $\sigma P(\boldsymbol{x})$ 取很大的正值,其作用迫使迭代点靠近可行域,因此解问题(10.6) 能够得到约束问题(10.2) 的近似解,而且 σ 越大,近似程度越好。通常将 $\sigma P(\boldsymbol{x})$ 称为罚项,σ 称为**罚因子**,$F(\boldsymbol{x},\sigma)$ 称为**罚函数**。

罚函数可有不同定义方法。$P(\boldsymbol{x})$的一般形式为

$$P(\boldsymbol{x})=\sum_{i=1}^{m}\Phi(g_i(\boldsymbol{x}))+\sum_{j=1}^{k}\Psi(h_j(\boldsymbol{x}))$$

Φ 和 Ψ 是满足下列条件的连续函数

当 $y\geqslant 0$ 时,$\Phi(y)=0$;

当 $y<0$ 时,$\Phi(y)>0$;

当 $y=0$ 时,$\Psi(y)=0$;

当 $y\neq 0$ 时,$\Psi(y)>0$

函数 Φ 和 Ψ 的典型取法如下

$$\Phi(g_i(\boldsymbol{x}))=[\max\{0,-g_i(\boldsymbol{x})\}]^{\alpha},\ \Psi(h_j(\boldsymbol{x}))=|h_j(\boldsymbol{x})|^{\beta}$$

其中 $\alpha\geqslant 1$, $\beta\geqslant 1$ 均为给定常数.通常取作 $\alpha=\beta=2$。

实际计算中,罚因子的选择很重要。如果 σ 太小,则罚函数的极小点远离约束问题的最优解;如果 σ 太大,则给计算增加困难。一般是取一个趋向无穷大的严格增正数列 $\{\sigma_k\}$,从 σ_1 开始,对每个 k 求解无约束问题 $\min f(\boldsymbol{x})+\sigma_k P(\boldsymbol{x})$,得到极小点的序列 $\{\bar{x}_{\sigma_k}\}$。在适当的条件下,这个序列收敛于约束问题的最优解。这样通过求解一系列无约束问题来获得约束问题最优解的方法称为序列无约束极小化方法,简称为 **SUMT 方法**。

(2) 计算步骤

(i) 给定初始点 $\boldsymbol{x}^{(0)}$,初始罚因子 σ_1,放大系数 $c>1$,允许误差 $\varepsilon>0$,置 $k=1$。

(ii) $\boldsymbol{x}^{(k-1)}$ 为初点,求解无约束问题 $\min f(\boldsymbol{x})+\sigma_k P(\boldsymbol{x})$,设其极小点为 $\boldsymbol{x}^{(k)}$。

(iii) 若 $\sigma_k P(\boldsymbol{x}^{(k)})<\varepsilon$,则停止计算,得到点 $\boldsymbol{x}^{(k)}$;否则令 $\sigma_{k+1}=c\sigma_k$,并置 $k=k+1$ 返回(ii)。

在序列无约束极小化过程中,$F(\boldsymbol{x}^{(k)},\sigma_k)$ 和 $f(\boldsymbol{x}^{(k)})$ 递增,$P(\boldsymbol{x}^{(k)})$ 递减。罚函数法也称**外点法**,所产生的序列在适当的条件下是收敛的。

实验十一　货车装货方案　整数规划

本实验的目的通过分析应用问题建立数学模型并求解，学习掌握整数线性规划及其求解的分枝定界算法；学习掌握 0-1 线性规划及其求解的隐枚举算法，并能运用 MATLAB 编程完成枚举和隐枚举求解。加强从"模型的建立到求解"的数学建模重要过程的实验活动。本实验的例题和习题包含了货车装货方案、投资场所的选定等问题。并给出了用 MATLAB 编写的分枝定界法、隐枚举法的参考程序。

§11.1　引例：货车装货方案

现有一节铁路货车，车箱长 10m，最大载重量为 40t(吨)，可以运载 7 类货物包装箱。包装箱的长度和重量不同，但宽和高相同且适合装车，每件包装箱不能拆开装卸，只能装或不装。每件货物的重量、长度与价值如表 11-1 所示。

表 11-1　货车装货数据表

货　物	长度(cm)	重量(t/件)	价值(千元)	件　数
1	55	0.5	40	8
2	58	1.7	37	8
3	62.4	3	58	6
4	49	2.2	36	7
5	40.6	3	35	3
6	53.3	1	45	4
7	66	4	50	8

请给出装货方案，使总的价值最大。

设 x_i 代表该车箱装入第 i 类货物包装箱的件数，则容易得到下列数学模型

$$\begin{cases} \max z = 40x_1 + 37x_2 + 58x_3 + 36x_4 + 35x_5 + 45x_6 + 50x_7 \\ s.t. \quad 55x_1 + 58x_2 + 62.4x_3 + 49x_4 + 40.6x_5 + 53.3x_6 + 66x_7 \leqslant 1000 \\ \qquad 0.5x_1 + 1.7x_2 + 3x_3 + 2.2x_4 + 3x_5 + x_6 + 4x_7 \leqslant 40 \\ \qquad x_1,\ x_2,\ x_7 \leqslant 8,\ x_3 \leqslant 6,\ x_4 \leqslant 7,\ x_5 \leqslant 3,\ x_6 \leqslant 4 \\ \qquad x_i \text{ 皆为正整数}, i = 1, 2, \cdots, 7 \end{cases}$$

此模型与前面所讲的线性规划模型非常相像，所不同的是在此要求变量是整数，此类优化问题称为**整数线性规划**(或**线性整数规划**)。对于变量限定只取 0 或 1 的整数线性规划，常称为 **0-1 线性规划**(或**线性 0-1 规划**)。

§11.2 整整线性线划基本理论复习

整数规划是一类要求变量取整数值的数学规划。若在整数规划中目标数和约束条件都是线性的,则称为整数线性规划(integer linear programming,简记 ILP);若要求变量只取 0 或 1 时,则称为 0-1 规划;若只要求部分变量取整数值,则称为混合整数规划。本实验主要内容是 ILP 和 0-1 线性规划(0-1 LP)。

1. 问题表述

在一般的线性规划中,增加限定:决策变量是整数,即为所谓 ILP 问题。其表述如下

$$\begin{cases} \min f = \boldsymbol{c}'\boldsymbol{x} \\ s.t.\quad A\boldsymbol{x} \leqslant (\text{或} =, \text{或} \geqslant) \boldsymbol{b} \\ \qquad x_j \geqslant 0 \ (j = 1, 2, \cdots, n) \\ \qquad x_j (j = 1, 2, \cdots, n) \text{ 取整数} \end{cases}$$

整数线性规划问题的标准形式为

$$\begin{cases} \min f = \boldsymbol{c}'\boldsymbol{x} \\ s.t.\quad A\boldsymbol{x} = \boldsymbol{b} \\ \qquad x_j \geqslant 0 \ (j = 1, 2, \cdots, n) \\ \qquad x_j (j = 1, 2, \cdots, n) \text{ 取整数} \end{cases}$$

其中 $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)$, $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)'$, $A = (a_{ij})_{m \times n}$, $\boldsymbol{b} = (b_1, b_2, \cdots, b_m)'$。

2. 算法——分枝定界法原理

求解 ILP 问题时,如果可行域是有界的,理论上可以用穷举法求解,对于变量不太多时此法可行,但当变量很多时这种穷举往往是行不通的。分枝定界法是 1960 年初由 Doig 和 Land 等人提出的可用于求解纯整数或混合整数线性规划问题的算法。分枝定界法比穷举法优越,它仅在一部分可行解的整数解中寻求最优解,计算量比穷举法小。当然若变量数目很大,其计算量也是相当可观的。

分枝定界法求解整数规划(最小化)问题的步骤为

初始,将要求解的整数规划问题称为 IL,将与它相应的线性规划问题称为问题 L。

(1) 解问题 L,可能得到以下情况之一

(i) L 没有可行解,这时 IL 也没有可行解,则停止。

(ii) L 有最优解,并且解变量都是整数,因而它也是 IL 的最优解,则停止。

(iii) L 有最优解,但不符合 IL 中的整数条件,此时记它的目标函数值为 f_0,这时若记 f 为 IL 的最优目标函数值,则必有 $f \geqslant f_0$。

(2) 迭代:第一步先分枝:在 L 的最优解中任选一个不符合整数条件的变量 x_j,设其值为 l_j,构造两个约束条件:$x_j \leqslant [l_j]$ 和 $x_j \geqslant [l_j]+1$,将这两个条件分别加入问题 L,将 L 分成两个后继问题 L_1 和 L_2。不考虑整数条件要求,求解 L_1 和 L_2。

然后定界:以每个后继子问题为一分枝并标明求解的结果,与其他问题的解的结果一道,找出最优目标函数值最小者作为新的下界,替换 f_0,从已符合整数条件的各分枝中,找

出目标数值为最小者作为新的上界 f^*，即有 $f^* \geqslant f \geqslant f_0$。

第二步 — 比较与剪枝：各分枝的最优目标函数中若有大于 f^* 者，则剪掉这一枝（即这一枝所代表的子问题已无继续分解的必要）；若小于 f^*，且不符合整数条件，则重复第一步骤，一直到最后得到最优目标函数值 $f = f^*$ 为止，从而得最优整态数解 x_j^*，$j = 1, 2, \cdots, n$。

下面用一个例子来说明上述过程。

例 1 求解下列整数规划

$$\begin{cases} \min f = 7x_1 + 3x_2 + 4x_3 \\ s.t. \quad x_1 + 2x_2 + 3x_3 \geqslant 8 \\ \qquad 3x_1 + x_2 + x_3 \geqslant 5 \\ \qquad x_j \geqslant 0 \ (j = 1, 2, 3) \\ \qquad x_1, x_2, x_3 \text{ 为整数} \end{cases}$$

解：放弃 x_1，x_2，x_3 为整数的条件求解线性规划问题 L 得：$\boldsymbol{x}^0 = (0.4, 3.8, 0)$，$f_0 = 14.2$；

按条件 $x_2 \leqslant 3$ 和 $x_2 \geqslant 4$ 将问题 L 分解成子问题 L_1 和 L_2 并赋它们下界为 14.2。

求解线性规划子问题 L_1 得：$\boldsymbol{x}^1 = (0.5, 3, 0.5)$，$f_1 = 14.5$；

求解线性规划子问题 L_2 得：$\boldsymbol{x}^2 = (1/3, 4, 0)$，$f_2 = 14.33$；$f_1 \wedge f_2 = 14.33$（$f_1$ 与 f_2 中最小者），由于 $f_1 \wedge f_2 = f_2$，而 $\boldsymbol{x}^2$ 中 $x_1 = 1/3$，因此以条件 $x_1 = 0$ 和 $x_1 \geqslant 1$ 将 L_2 分成两个子问题 L_3 和 L_4 并赋它们下界为 14.33。

求解线性规划子问题 L_3，得：$\boldsymbol{x}^3 = (0, 5, 0)$，$f_3 = 15$；

求解线性规划子问题 L_4，得：$\boldsymbol{x}^4 = (1, 4, 0)$，$f_4 = 15$；

由于 $\boldsymbol{x}^3$ 和 $\boldsymbol{x}^4$ 是原整数规划问题的可行解且 $f_3 \wedge f_4 = 15$，所以置 $f^* = 15$ 作为上界。

以下再将 L_1 分枝，因 $x_1^1 = 0.5$，所以可按条件 $x_1 = 0$ 和 $x_1 \geqslant 1$ 将 L_1 分成两个问题 L_5 和 L_6，并赋予它们下界 14.33。求解线性规划子问题 L_5 得：$\boldsymbol{x}^5 = (0, 3, 2)$，$f_5 = 17$；求解线性规划于问题 L_6 得：$\boldsymbol{x}^6 = (1, 0, 7/3)$，$f_4 = 16.33$；

由于 f_5、$f_6 > f_3 \wedge f_4 = 15$，所以 L_5 和 L_6 都没有继续分枝求解的必要，至此求得最优解为 $\boldsymbol{x}^* = \boldsymbol{x}^3 = (0, 5, 0)$ 及 $\boldsymbol{x}^* = \boldsymbol{x}^4 = (1, 4, 0)$，最优目标函数值为 $f = f_3 = f_4 = 15$。

上述求解过程可用下两的框图 11-1 表示。

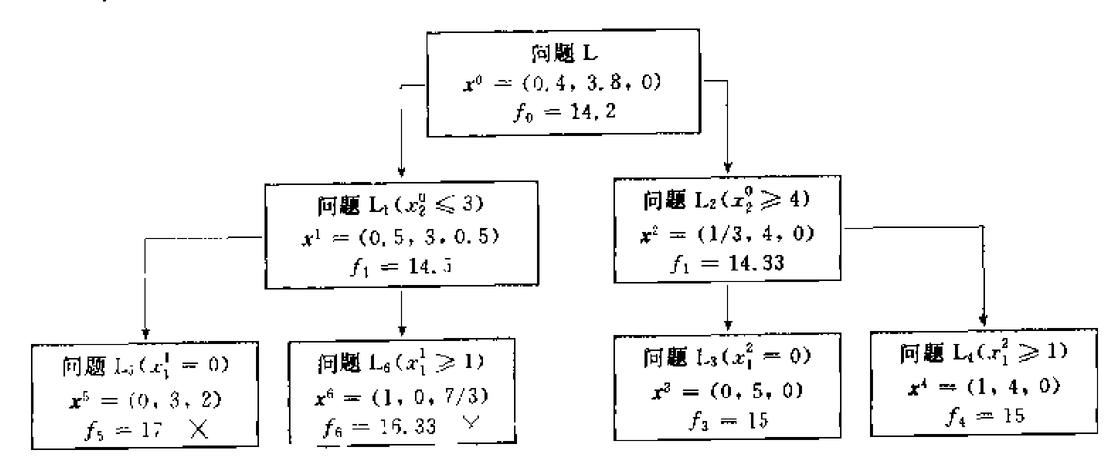


图 11-1 分枝定界法的框图

§ 11.3 整数数性规划分枝定界法 MATLAB 程序

%M 函数 ILp. m

```
    function  [x, f] = ILp (c, A, b, vlb, vub, x0, neqcstr, pre)
%   [x, f] = ILp (c, A, b, vlb, vub, x0, neqcstr, pre)求解下列
%          整数线性规划问题
%     min  f = c'* x, s.t. A* x( = b, vlb( = x( = vub
%        x 的分量全为整数
%   其中 x0 是初始值,可以用[]代替 x0; neqcstr 表示约束条件
%   Ax≤b 中的前 neqcstr 个是等式,neqcstr=0 时可以省略,
%   此时也可以省略 x0,即调用格式:x=ILp (c, A, b, vlb, vub)
%   求解整数线性规划。pre 是精度。
%   返回结果 x 是最优解,f 是最优解处的函数值。
if nargin(8, pre = 0;
  if nargin(7, neqcstr = 0;
    if nargin(6, x0 = [];
      if nargin(5, vub = [];
        if nargin(4, vlb = [];
        end, end, end, end, end
% Set to column vectors
x0 = x0(:); c = c(:); b = b(:); vlb = vlb(:); vub = vub(:);
mm = 1; j = 1; nvars = length(c');
fvub = inf; xall = []; fall = []; x _ f _ b = [];
[xtemp ztemp, how] = lp (c, A, b, vlb, vub, x0, neqcstr, -1);
ftemp = c'* xtemp;
if strcmp (bow, 'ok'),
    temp0 = round(xtemp); temp1 = floor(xtemp);
    temp2 = find(abs(xtemp - temp0))pre);
    mtemp = lengtb(temp2);
    if ~isempty(temp2)
      x _ f _ b = [xtemp; ftemp; vlb; vub];
      while j( = mm
        i = 1;
        wbile i( = mtemp
          if x _ f _ b(nvars + 1, j)( = fvub
          vlbl = x _ f _ b(nvars + 2:2* nvars + 1, j);
          vubl = x _ f _ b(2* nvars + 2:3* nvars + 1, j);
          vubl(temp2 (i)) = tenpl(temp2(i));
```

```
        [xtemp z, how] = lp(c, [A; c'], [b; fvub],...
                    vlbl, vubl, x0, neqcstr, -1);
        ftemp = c'* xtemp;
        if strcmp(how, 'ok'),
        templ0 = round(xtemp); templ1 = floor(xtemp);
        templ2 = find(abs(xtemp - templ0)>pre);
        if isempty(templ2),
          xall = [xall, xtemp]; fall = [fall, ftemp];
          fvub = min([fvub, fall]);
        elseif ftemp<=fvub
          x_f_b = [x_f_b, [xtemp; ftemp; vlbl; vubl]];
        end, end, end
      if x_f_b(nvars + 1, j)<=fvub
        vlbr = x_f_b(nvars + 2:2* nvars + 1, j);
        vlbr(temp2(i)) = temp1(temp2(i)) + 1;
      vubr = x_f_b(2* nvars + 2:3* nvars + 1, j);
      [xtemp z, how] = lp (c, [A; c'], [b; fvub], vlbr,...
                          vubr, x0, neqcstr, -1);
      ftemp = c'* xtemp;
      if strcmp(bow, 'ok'),
        tempr0 = round(xtemp); tempr1 = floor(xtemp);
        tempr2 = find(abs(xtemp - tempr0)>pre);
        if isempty(tenpr2),
          xall = [xall, xtemp]; fall = [fall, ftemp];
          fvub = min([fvub, fall]);
        elseif ftemp<=fvub
          x_f_b = [x_f_b, [xtemp; ftemp; vlbr; vubr]];
        end, end, end
      i = i + 1; end
    xint = x_f_b(1:nvars,:); [m, mm] = size(xint); j = j + 1;
    if j>mm, break, end
    temp0 = round(xint(:, j)); temp1 = floor(xint(:,j));
    temp2 = find(abs(xint(:, j) - temp0)>pre);
    mtemp = lengtb(temp2); end,
else, x = xtemp; f = ftemp; end,
if ~isempty(fall)
  fmin = min(fall); nmin = find(fall == fmin);
  x = xall(:, nmin); f = fmin; end,
else, x = nan* ones(1, nvars); end
```

§11.4 0-1 型整整线性线划

1. 0-1 型整数线性规划的提法

0-1 型整数线性规划是一类特殊的整数规划，它的变量仅取值 0 或 1；它的提法如下

$$\begin{cases}\min f = c'x \\ s.t. \quad Ax = b \\ \qquad x_j (j = 1, 2, \cdots, n) \text{ 取 } 0 \text{ 或 } 1\end{cases}$$

其中 $c = (c_1, c_2, \cdots, c_n)$，$x = (x_1, x_2, \cdots, x_n)'$，$A = (a_{ij})_{m \times n}$，$b = (b_1, b_2, \cdots, b_m)'$。

我们称此时的决策变量为 **0-1 变量**，或称**二进制变量**。在实际问题中，如果引进 0-1 变量，就可以把各种需要分别讨论的线性（或非线性）规划问题统一在一个问题中讨论了。

2. 求解 0-1 线性规划的隐枚举法

求解整数线性规划的分枝定界法也是一种隐枚举法，0-1 规划可以通过增加限定 $0 \leqslant x_i \leqslant 1$ 的整数规划来求解。

在此主要介绍一种针对 0-1 线性规划特点的隐枚举法算法。所谓隐枚举，是一种“聪明”的枚举，通过设计一些方法，检查变量组合的一部分，而不必全部检查 n 个变量的 2^n 个取值组合。要说明的是，对有些问题（特别是对于一部分决策变量是 0-1 变量的混合线性规划）隐枚举法有时难以适用，所以穷举法还是必要的。

隐枚举法原题与算法步骤

(i) 记 $f_0 = \infty$，将 n 个决策变量构成的 x 的可能的 2^n 种取值组合按二进制（或某种顺序）排序；

(ii) 按上述顺序对 x 的取值首先检测 $f = c'x \leqslant f_0$ 是否成立，若不成立则放弃该取值的 x，按次序换(i)中下一组 x 的取值重复上述过程；若成立，则转下步；

(iii) 对 x 逐一检测 $Ax \leqslant b$ 中的 m 个条件是否满足，一旦检测某条件不满足便停止检测后面的条件，而放弃这一组 x，按次序换(i)中下一组 x 的取值执行(ii)；若 m 个条件全满足，则转下步；

(iv) 记 $f_0 = \min(f_0, f)$，按次序换(i)中下一组 x 的取值，执行(ii)；

(v) 最后一组满足 $f = c'x \leqslant f_0$ 和 $Ax \leqslant b$ 的 x 即为最优解。

注：在执行上述算法步骤时，可以及时地记录所有满足 $f = c'x = f^*$（f^* 为最优值）的 x，以便求所有最优解。

§11.5 0-1 型整整线性线划计算的 MATLAB 程序

在下面的枚举和隐枚举程序中用到命令 B = de2bi(D)，其作用是将十进制数向量 D 转换为相应二进制数按位构成的以 0，1 为元素的矩阵 B。

%M 函数 de2bi.m

```
function b = de2bi(d, n, p)
% DE2BI    转换十进制数为二进制数。
```

```
%           B = DE2BI(D)转换正整数向量 D 成二进制矩阵 B。
%           二进制矩阵 B 的每一行表示十进制向量 D 中相应的数。
%           B = DE2BI(D, N)转换正整数向量 D 成二进制矩阵 B,
%           但指定 B 的列数为 N。
%           B = DE2BI(D, N, P)转换正整数向量 D 成 p 进制矩阵 B。
%           p 进制矩阵 B 的每一行表示十进制向量 D 中相应的数。
d = d(:); len_d = length(d);
if min(d)<0, error('Cannot convert a negative number');
elseif ~isempty(find(d == inf)),
    error('Input must not be Inf.');
elseif find(d ~ = floor(d)),
    error('Input must be an integer.');
end;
if nargin<2,
  tmp = max(d); b1 = [];
  while tmp>0
    b1 = [b1 rem(tmp, 2)]; tmp = floor(tmp/2);
  end;
  n = length(b1);
end;
if nargin<3, p = 2; end;
b = zeros(len_d, n);
for i = 1:len_d
  j = 1; tmp = d(i);
  while (j< = n)&(tmp>0)
    b(i, j) = rem(tmp, p); tmp = floor(tmp/p);
    j = j + 1;
end; end;
```

%M 函数 L01p_e.m(枚举法)

```
function [x, f] = L01p_e(c, A, b, N)
% [x, f] = L01p_e(c, A, b, N)用枚举法求解下列
%       0-1 线性规划问题
%       min f = c'* x, s.t. A* x< = b, x 的分量全为整数 0 或 1,
%   其中 N 表示约束条件 Ax≤b 中的前 N 个是等式,N = 0 时可以省略。
%   返回结果 x 是最优解,f 是最优解处的函数值。
if nargin<4, N = 0; end
c = c(:); b = b(:);
[m, n] = size(A); x = []; f = abs(c')* ones(n, 1); i = 1;
while i< = 2^n
```

```
    B = de2bi(i - 1, n)';
    t = A* B - b; t11 = find(t(1:N, :) ~ = 0);
    t12 = find(t(N + 1:m,:))0); t1 = [t11; t12];
    if isempty(t1)
        f = min([f, c'* B]);
        if c'* B = = f, x = B; end
    end
    i = i + 1;
end
```

%M 函数 L01p _ ie

```
function [x, f] = L01p _ ie(c, A, b, N)
%  [x, f] = L01p _ ie(c, A, b, N)用隐枚举法求解下列 0-1 线性规划问题
%       min f = c'* x, s.t. A* x( = b, x 的分量全为整数 0 或 1,
%  其中 N 表示约束条件 Ax≤b 中的前 N 个是等式,N = 0 时可以省略。
%  返回结果 x 是最优解,f 是最优解处的函数值。
if nargin(4, N = 0; end
c = c(:); b = b(:);
A = [ - A(1:N,:); A]; b = [ - b(1:N); b];
[m, n] = size(A); x = []; f = abs(c')* ones(n, 1);
A = [c'; A]; b = [f; b]; i = 1;
while i( = 2^n
    B = de2bi(i - 1, n)';
    j = 1; t1 = A(j,:)* B - b(j);
    while (t1( = 0&j(m + 1)
        j = j + 1; t1 = A(j,:)* B - b(j);
        if t1)0, j = 1; end;
    end
    if j = = m + 1
        x = B; f = c'* B;
        b(1) = min([b(1), f]);
    end
    i = i + 1;
end
```

§11.6 实 验 例 题

例 2 (货车装货方案)调用命令 ILp.m 计算如下

```
》clear; f = -[40; 37; 58; 36; 35; 45; 50]; % 转换求 max 为求 min
》a = [55, 58, 62.4, 49, 40.6, 53.3, 66; 0.5, 1.7, 3, 2.2, 3, 1, 4];
```

```
》b = [1000; 40]; lb = zeros(7, 1); ub = [8; 8; 6; 7; 3; 4; 8];
》tic; [x, f] = ILp(f, a, b, lb, ub, [], 0, 0.001), toc
x=        %输出
      3
      1
      6
      0
      3
      4
      1
f=
  -840
elapsed_time=
    328.84
```

注:分枝定界法本质上还是一种枚举法,并不是一种完全有效的算法,当变量较多或条件较复杂时,计算耗时可能很多。本例用时328.84秒。

例3 某厂拟用集装箱托运甲乙两种货物,每箱的体积、重量、可获利润以及托运所受限制如表11-2。

表 11-2

货　物	体　积	重　量	利　润
	每箱(m^3)	每箱(t)	每箱(千元)
甲	5	2	20
乙	4	5	10
托运限制	24	13	

问两种货物各托运多少箱,可使获得利润为最大?

解:这个问题可用整数规划求解。设 x_1 和 x_2 分别为甲和乙两种货物的托运箱数(皆为非负整数),则有

$$\begin{cases} \max\ f = 20x_1 + 10x_2 \\ s.t.\quad 5x_1 + 4x_2 \leqslant 24 \\ \qquad 2x_1 + 5x_2 \leqslant 13 \\ \qquad x_j \geqslant 0\ (j = 1,\ 2) \\ \qquad x_1,\ x_2 \text{ 为整数} \end{cases}$$

首先放弃整数性条件,求线性规划问题

```
》c = [-20, -10]; a = [5, 4; 2, 5]; b = [24; 13];      %转换 max 为 min
》[x, f] = linprog(c, a, b, [], [], [0; 0], [])
x=
```

```
    4.8000
         0
f=
    -96
》c=[-20, -10]; a=[5, 4; 2, 5]; b=[24; 13];      %增加条件 x1≤4
》[x, f]=linprog(c, a, b, [], [], [0; 0], [4; inf])
x=
    4
    1
f=
    -90
》c=[-20, -13]; a=[5, 4; 2. 5]; b=[24; 13];      %增加条件 x1≥5
》x=linprog (c, a, b, [], [], [5; 0], [])
无可行解
```

可见 $x=(4, 1)^T$，$f=-90$ $(-f=90)$，问题得以解决。

调用 ILp.m 求解：

```
》c=[-20, -10]; a=[5, 4; 2, 5]; b=[24; 13];
》[x, f]=ILp(c, a, b, [0; 0], [inf; inf], [], 0, 0.0001)
x=
          4
          1
f=
        -90
```

例 4　求解下列整数规划

$$\begin{cases}\min f = x_1 + x_2 - 4x_3 \\ s.t. \quad x_1 + x_2 + 2x_3 \leqslant 9 \\ \qquad x_1 + x_2 - x_3 \leqslant 2 \\ \qquad -x_1 + x_2 + x_3 \leqslant 4 \\ \qquad x_j \geqslant 0 \ (j = 1, 2) \\ \qquad x_1, x_2 \text{ 为整数}\end{cases}$$

解：调用 ILp.m 求解

```
》c=[1, 1, -4]; a=[1, 1, 2; 1, 1, -1; -1, 1, 1]; b=[9; 2; 4];
》[x, f]=ILp(c, a, b, [0; 0; 0], [inf; inf; inf], [], 0, 0.0001)
x=        %有两枝得到同样解
           0       0
           0       0
           4       4
f=
```

−16

例 5　求解下列 0-1 型整数规划

$$\begin{cases}\max f = -3x_1 + 2x_2 - 5x_3 \\ s.t. \quad x_1 + 2x_2 - x_3 \leqslant 2 \\ \qquad x_1 + 4x_2 + x_3 \leqslant 4 \\ \qquad x_1 + x_2 \leqslant 3 \\ \qquad 4x_2 + x_3 \leqslant 6 \\ \qquad x_1, x_2, x_3 \text{ 为 } 0 \text{ 或 } 1\end{cases}$$

解：分别采用枚举法和隐枚举法求解如下

```
》c=[3, -2, 5];      %转换求 max 为求 min
》a=[1, 2, -1; 1, 4, 1; 1, 1, 0; 0, 4, 1]; b=[2; 4; 3; 6];
》x1=L01p_e(c, a, b); x2=L01p_ie(c, a, b); x1, x2
x1=
      0
      1
      0
x2=
      0
      1
      0
```

即得问题的解。

例 6　一架货运飞机，有效载重量为 24t(吨)，可运输物品的重量及运费收入如表 11-3 所示，其中各物品只有一件可供选择。

表 11-3

物　品	1	2	3	4	5	6
重量(t)	8	13	6	9	5	7
收入(万元)	3	5	2	4	2	3

问如何选运物品运费总收入最多？

解：记 $x_i = \begin{cases}1 & \text{当选运第 } i \text{ 种物品时} \\ 0 & \text{其他}\end{cases}$

则有

$$\begin{cases}\max f = 3x_1 + 5x_2 + 2x_3 + 4x_4 + 2x_5 + 3x_6 \\ s.t. \quad 8x_1 + 13x_2 + 6x_3 + 9x_4 + 5x_5 + 7x_6 \leqslant 24 \\ \qquad x_1, x_2, x_3, x_4, x_5, x_6 \text{ 为 } 0 \text{ 或 } 1\end{cases}$$

采用枚举和隐枚举法计算得(将题中的求 max 转换为求 min)：

```
》c=-[3, 5, 2, 4, 2, 3]; a=[8, 13, 6, 9, 5, 7]; b=24;
```

```
》x1 = L01p _ e(c, a, b); x2 = L01p _ ie(c, a, b); x1 _ x2 = [x1, x2]
x1 _ x2 =
        1      1
        0      0
        0      0
        1      1
        0      0
        1      1
```

可见分别选运第 1，4 和 6 种物品。

§11.7 实验习题

1 求解下列整数规划问题

$$\begin{cases} \max f = 3x_1 - x_2 \\ s.t. \quad 3x_1 - 2x_2 \leqslant 3 \\ \qquad 5x_1 + 4x_2 \geqslant 10 \\ \qquad 2x_1 + x_2 \leqslant 5 \\ \qquad x_1, x_2 \geqslant 0 \\ \qquad x_1, x_2 \text{ 为整数} \end{cases}$$

2 （生产计划制定）某工厂用甲、乙两种原料生产 A，B，C，D 四种产品，每种产品消耗原料定额如表 11-4 所示，现有甲原料 18t(吨)，乙原料 3t。请制定一个使总利润最大的生产计划。

表 11-4

	A(万件)	B(万件)	C(万件)	D(万件)
甲 t	2	3	10	4
乙 t	—	—	2	0.5
单位利润(万元/万件)	9	8	50	19

3 某机械厂制造 A、B 和 C 三种机床，每种机床须用不同数量的两类电气部件：部件 1 和部件 2。设机床 A、B 和 C 各用部件 1 的个数分别为 4、6 和 2，各用部件 2 的个数分别为 4、3 和 5；在任何一个月内共有 22 个部件 1 和 25 个部件 2 可用；生产 A，B 和 C 三种机床每台的利润分别为 5 万元、6 万元和 4 万元。问 A，B 和 C 三种机床每月各生产几台，才能使厂部取得最大利润。

4 （投资场所选定）某公司拟在市东、西、南三区建立门市部。拟议中有 7 个位置点 A_i $(i = 1, 2, \cdots, 7)$ 可供选择。规定：在东区，由三个点 A_1、A_2、A_3 中至多选两个；在西区，由两个点 A_4、A_5 中至少选一个；在南区，由两个点 A_6、A_7 中至少选一个。投资总额不能超过 700 万元。设备投资费与每年可获利润见表 11-5。问应选择哪几个点可使年利润为

最大？

表 11-5

	A_1	A_2	A_3	A_4	A_5	A_6	A_7
设备投资费(万元)	13	18	21	29	11	28	19
年终获利润(万元)	21	25	27	37	19	33	25

5 求解 0-1 规划

$$\begin{cases} \min f = 4x_1 + 3x_2 + 2x_3 \\ s.t. \quad 2x_1 - 5x_2 + 3x_3 \leqslant 4 \\ \qquad 4x_1 + x_2 + 3x_3 \geqslant 3 \\ \qquad x_2 + x_3 \geqslant 1 \\ \qquad x_1,\ x_2,\ x_3 \text{ 为 } 0 \text{ 或 } 1 \end{cases}$$

6 求解 0-1 规划

$$\begin{cases} \max f = -2x_1 - 5x_2 - 3x_3 - 4x_4 \\ s.t. \quad -4x_1 + x_2 + x_3 + x_4 \geqslant 0 \\ \qquad -2x_1 + 4x_2 + 2x_3 + 4x_4 \geqslant 4 \\ \qquad x_1 + x_2 - x_3 + x_4 \geqslant 1 \\ \qquad x_1,\ x_2,\ x_3 \text{ 为 } 0 \text{ 或 } 1 \end{cases}$$

实验十二　通信联络网的建立
图与网络优化

本实验的目的是通过对应用问题的介绍及求解，学习图与网络的相关知识及其应用方法，理解并掌握最优树及最短路径的Kruskal、Dijkstra等算法，并能运用MATLAB或其他算法语言编写算法程序，强化用图与网络建立解决应用问题的模型并求解的实验活动。本实验研究了建立通信联络网和设备更新方案等应用问题，并给出了用MATLAB编写的Kruskal和Dijkstra算法程序。

§12.1　引例:通信联络网的建立

设有6个居民点(如图12-1(a)所示)，每条边代表两居民点的道路，数字代表路长。现要在这6个居民点之间设置通信线路两，以保证6个居民点的联络。如果已知设置通信线路代价与道路长成正比，问如何建立该通信联络网，而使联网代价最小。

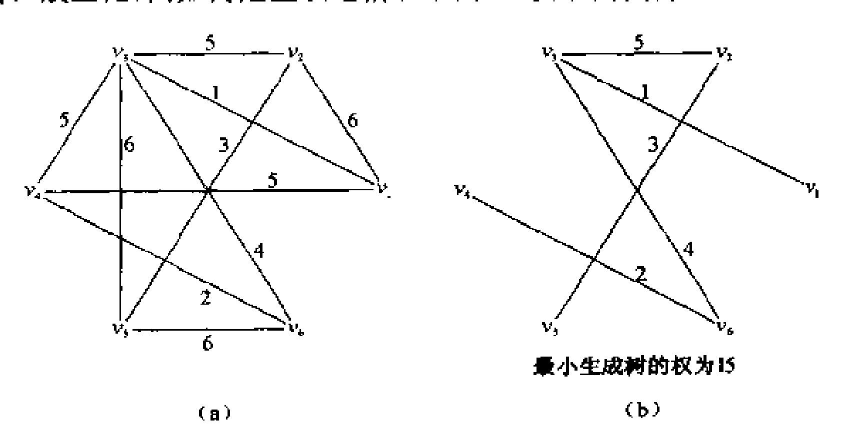


图12-1　通信联络网

图12-1(b)是一种建立通信联络网的方案。如果在 v_1 和 v_2 间再连一根线路，则 v_1、v_2 和 v_3 形成了一个闭合的线路，称为回路。但由于 v_1 已经通过 v_3 与 v_2 建立了线路连通，所以再连 v_1 和 v_2 是多余的且增加成本。如此看来最佳联网方案中不允许有回路。这种没有回路的连通的图，称为树，包含了原图所有顶点的树称为原图的**生成树**，联网成本最小的生成树称为**最小生成树(最优树)**。对本例而言是寻找最小生成树。

§12.2　建与网络的基本建论复习

1. 基本概念和名词

由若干个不同的点(称之为**复点**或**节点**)与其中某些顶点的连线所组成的图形称为**图**，

如图 12-1 中 v_i 为顶点。

如果图中的每条边都有一个具体的数与之对应，称这些数为权，称这样的图为**赋权图**或**网络**，如图 12-1(a)所示。

图与网络的基本概念如下

(1) **边与弧**：把两点之间不带箭头的连线称为**边**，带箭头的连线（例如图 12-2 所示）称为**弧**。

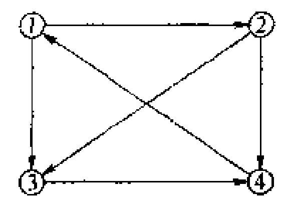


图 12-2 有向图

(2) **无向图与有向图**：如果一个图 G 是由顶点和边构成的，称之为**无向图**（如图 12-1），记为 $G=(V,E)$，V 和 E 分别是 G 的顶点的集合和边的集合，$V=\{v_1, v_2, \cdots, v_n\}$，$E=\{e_1, e_2, \cdots, e_m\}$。如果一个图 G 是由顶点和弧构成的，称之为**有向图**（如图 12-2），记为 $G=(V,A)$，其中 V 和 A 分别是 G 的顶点的集合和弧的集合，$A=\{a_1, a_2, \cdots, a_m\}$。通常用顶点的有序对（对无向图是无序对）$(v_i, v_j)$来表示弧或边，如图 12-1(b)，$G=(V,E)$，$V=\{v_1, v_2, v_3, v_4, v_5, v_6\}$，$E=\{(v_1, v_3), (v_2, v_3), (v_2, v_5), (v_3, v_6), (v_1, v_6)\}$。

(3) **链**：在图 $G=(V,E)$ 中，点与边的交错序列 $(v_i^1, e_i^1, v_i^2, e_i^2, \cdots, v_i^{k-1}, e_i^{k-1}, v_i^k)$，其中 e_i^t 为连结 v_i^t 和 v_i^{t+1} 的边$(t=1, 2, \cdots, k-1)$，称为一条连结 v_i^1 和 v_i^k 的链，如图 12-2，$\{(1, 4), (4, 2), (2, 3)\}$为一条链。

(4) **路径**：如果 $(v_i^1, a_i^1, v_i^2, a_i^2, \cdots, v_i^{k-1}, a_i^{k-1}, v_i^k)$ 是图 G 中一条链，并且 a_i^t 为从 v_i^t 指向 v_i^{t+1} 的弧$(t=1, 2, \cdots, k-1)$，称之为从 v_i^1 到 v_i^k 的路径。如图 12-2，$\{(1, 2), (2, 3), (3, 4)\}$为一条路径。

(5) **回路**：闭合的路径称为**回路**。

(6) **圈**：闭合的链称为**圈**。对无向图而言，链与路径、回路与圈是同一的。

(7) **连通图**：图 G 中任何两个点之间至少有一条链，称 G 为**连通图**。

(8) **树与生成树**：一个无圈的连通图称为**树**。若 $G_1=(V_1, E_1)$ 是连通图 $G_2=(V_2, E_2)$ 的生成子图（即 $V_1=V_2$，$E_1 \subseteq E_2$），且 G_1 本身是树，则称 G_1 为 G_2 的**生成树**。

(9) **邻接矩阵**：以 b_{ij} 表示图 G 中从顶点 v_i 到 v_j 的弧的数目（无向图只考虑 v_i 与 v_j 间的边的数目），则矩阵 $B=(b_{ij})$ 称为图 G 的**邻接矩阵**。

(10) **带权邻接矩阵**：以 w_{ij} 表示网络 G 中从顶点 v_i 到 v_j 的弧的权（无向网只考虑 v_i 与 v_j 间的边的权），当 v_i 到 v_j 无弧或边时，$w_{ij}=\infty$，则矩阵 $W=(w_{ij})$ 称为图 G 的**带权邻接矩阵**。如图 12-4 的带权邻接矩阵为（这里是用 MATLAB 输出结果的，Inf 表示∞）

```
W=
    Inf    Inf    10     Inf    30     100
    Inf    Inf    5      Inf    Inf    Inf
    Inf    Inf    Inf    50     Inf    Inf
    Inf    Inf    Inf    Inf    Inf    10
    Inf    Inf    Inf    20     Inf    60
    Inf    Inf    Inf    Inf    Inf    Inf
```

2. 最小生成树与 Kruskal 算法

树是一类特殊的图。1847 年 Kirchhoff 在研究电网络时，便发展了有关树的理论。树

在分子结构、电网络分析、计算机科学等领域有广泛应用。

最小生成树:在赋权图 G 中,求一棵生成树,使其总权最小,称这棵生成树为图 G 的最小生成树。

Kruskal 算法思想及步骤:Kruskal(1959)提出了求图的最小生成树的算法,其中心思想是每次添加权尽量小的边,使新的图无圈,直到生成一棵树为止,便得最小生成树。其算法步骤如下

(i) 把赋权图 G 中的所有边按照权的非减次序排列。

(ii) 按(i)排列的次序检查 G 中的每一条边,如果这条边与已得到的边不产生圈,则取这一条边为解的一部分。

(iii) 若已取到 $n-1$ 条边,算法终止。此时以 V 为顶点集,以取到的 $n-1$ 条边为边集的图即为最小生成树。

3. 最短路径与 Dijkstra 算法

最短路径问题是图论中的一个基本问题。它在通讯、石油管线铺设、公路网等实际问题中有着广泛的应用。

最短路径问题:在赋权有向图 G 中,求一条总权最小的 v_i 至 v_j 的**路径**的问题,就是最短路径问题。

Dijkstra 算法的基本思想:如果 $v_1, v_2, \cdots, v_i, \cdots, v_j, \cdots, v_n$ 是某图G 从 v_1 到 v_n 的最短路径,则它的子路 $v_i, \cdots, v_j$ 一定是从 v_i 到 v_j 的最短路径。

Dijkstra 算法的步骤:该算法可求得网络中从某顶点到其他所有顶点的最短路径,算法步骤为

(i) 假设网络 G 有 n 个顶点,用带权的邻接矩阵 W 来表示,$W(i, j)$表示从顶点 v_i 到v_j 的弧或边上的权值,不存在弧或边的权值用∞(在 MATLAB 中为 Inf)表示。S 为已求出的从已知始点 v_i 出发的最短路径的终点的集合,它的初始状态为空集。则从 v_i 出发到图上其余各顶点 v_k 可能达到的最短路径长度的初值为:$D(k) = \min\{W(i, k) \mid v_k \in V - \{i\}\}$;

(ii) 选择 v_j,使得:$D(j) = \min\{D(k) \mid v_k \in V - S\}$,$v_j$ 就是当前求得的一条从始点 v_i 出发的最短路径的终点。令 $S = S \cup \{j\}$;

(iii) 修改从 v_i 出发到集合 $V-S$ 上任一顶点 v_k 可达的最短路径长度。如果 $D(j) + W(j, k) < D(k)$,则修改 $D(k)$为:$D(k) = D(j) + W(j, k)$;

(iv) 重复操作(ii)、(iii)共 $n-1$ 次,并记录各最短路径经过的所有顶点。由此得到从始点 v_i 到图上其余各顶点的最短路径是依路径长度递增的序列。

§12.3 Kruskal 算法与 Dijkstra 算法的 MATLAB 程序

1. Kruskal 算法的 MATLAB 程序

```
                        %M 函数 mintreek.m
function [Wt, Pp] = mintreek(n, W)
%图论中最小生成树 Kruskal 算法及画图程序 M 函数
%格式 [Wt, Pp] = mintreek(n, W);n 为图顶点数,W 为图的带权邻接矩
%        阵,不构成边的两顶点之间的权用 inf 表示。显示最小生成树的边及
```

```
%          顶点,Wt 为最小生成树的权,Pp(:, 1:2)为最小生成树边的两顶点,
%          Pp(:, 3)为最小生成树的边权,Pp(:, 4)为最小生成树边的序号;
%附图,红色连线为最小生成树的图;
tmpa = find(W~ = inf); [tmpb, tmpc] = find(W~ = inf);
w = W(tmpa); e = [tmpb, tmpc]; %w 是 W 中非 inf 元素按列构成的向量
                              %e 的每一行元素表示一条边的两个顶点的序号
[wa, wb] = sort(w); E = [e(wb, :), wa, wb]; [nE, mE] = size(E);
temp = find(E(:, 1) - E(:, 2)); E = E(temp, :);
P = E(1, :); k = length(E(:, 1));
while (rank(E)>0)
  temp1 = max(E(1, 2), E(1, 1)); temp2 = min(E(1, 2), E(1, 1));
  for i = 1:k;
    if (E(i, 1) == temp1), E(i, 1) = temp2; end;
    if (E(i, 2) == temp1), E(i, 2) = temp2; end
  end;
  a = find(E(:, 1) - E(:, 2)); E = E(a, :);
  if (rank(E)>0), P = [P; E(1, :)]; k = length(E(:, 1)); end;
  end;
  Wt = sum(P(:, 3)); Pp = [e(P(:, 4), :), P(:, 3:4)];
  for i = 1:length(P(:, 3)); %显示顶点 v_i 与边 e_j
  disp(['    ','e', num2str(P(i, 4)), '  ','(v',...
  num2str(P(i, 1)), '  ','v', num2str(P(i, 2)), ')']);
  end;
  %以下是画图程序
  axis equal; hold on
  [x, y] = cylinder(1, n); xm = min(x(1, :)); ym = min(y(1, :));
  xx = max(x(1, :)); yy = max(y(1, :));
  axis([xm - abs(xm)*0.15, xx + abs(xx)*0.15, ym - abs(ym)*0.15, ...
  yy + abs(yy)*0.15]); plot(x(1, :), y(1, :), 'ko')
  for i = 1:n; temp = ['  v', int2str(i)];
      text(x(1, i), y(1, i), temp); end;
  for i = 1:nE, plot(x(1, e(i, :)), y(1, e(i, :)), 'b'); end;
  for i = 1:length(P(:, 4)),
    plot(x(1, Pp(i, 1:2)), y(1, Pp(i, 1:2)), 'r'); end;
  text(-0.35, -1.2, ['最小生成树的权为','  ', num2str(Wt)]);
  title('红色连线为最小生成树'); axis('off'); hold off
```

2. Dijkstra 算法的 MATLAB 程序

```
                         %M 函数 minroute.m
function [S, D] = minroute(i, m, W)
```

```
% 图与网络论中求最短路径的 Dijkstra 算法 M 函数
% 格式      [S, D] = minroute(i, n, W)
%           i 为最短路径的起始点,m 为图顶点数,W 为图的带权邻接矩阵,
%           不构成边的两顶点之间的权用 inf 表示。显示结果为:S 的每
%           一列从上到下记录了从始点到终点的最短路径所经顶点的序号;
%           D 是一行向量,记录了 S 中所示路径的大小;
dd = []; tt = []; ss = []; ss(1, 1) = i; V = 1:m; V(i) = []; dd = [0; i];
%   dd 的第二行是每次求出的最短路径的终点,第一行是最短路径的值
kk = 2; [mdd, ndd] = size(dd);
while ~isempty(V)
    [tmpd, j] = min(W(i, V)); tmpj = V(j);
    for k = 2:ndd
        [tmp1, jj] = min(dd(1, k) + W(dd(2, k), V));
        tmp2 = V(jj); tt(k - 1, :) = [tmp1, tmp2, jj];
    end
    tmp = [tmpd, tmpj, j; tt]; [tmp3, tmp4] = min(tmp(:, 1));
    if tmp3 == tmpd, ss(1:2, kk) = [i; tmp(tmp4, 2)];
    else, tmp5 = find(ss(:, tmp4)~ = 0); tmp6 = length(tmp5);
        if dd(2, tmp4) == ss(tmp6, tmp4)
            ss(1:tmp6 + 1, kk) = [ss(tmp5, tmp4); tmp(tmp4, 2)];
            else, ss(1:3, kk) = [i; dd(2, tmp4); tmp(tmp4, 2)];
    end; end
    dd = [dd, [tmp3; tmp(tmp4, 2)]]; V(tmp(tmp4, 3)) = [];
    [mdd, ndd] = size(dd); kk = kk + 1;
end; S = ss; D = dd(1, :);
```

§12.4 实验例题

例 1 (通信联络网的建立)运用 Kruskal 算法求出图 12-1(a)的最小生成树,即可得到最佳建立网络的方案。求解如下

```
》clear; n = 6; w = inf* ones(6);
》w(1, [2, 3, 4]) = [6, 1, 5]; w(2, [3, 5]) = [5, 3];
》w(3, [4, 5, 6]) = [5, 6, 4]; w(4, 6) = 2; w(5, 6) = 6;
》[a, b] = mintreek(n, w)
    e2(v1 v3)      %输出顶点与边的标记,最小树的构成一目了然
    e9(v4 v6)
    e6(v2 v5)
    e8(v1 v4)
    e3(v2 v1)
```

```
a=
    15         %最小生成树的权值
b=             %其含义见程序 mintreek.m 的说明
    1     3     1     2
    4     6     2     9
    2     5     3     6
    3     6     4     8
    2     3     5     3
```

这里略去了所附图形，该图是图 12-1(a)与 12-1(b)的重叠。

例 2 （设备更新问题）某公司使用一种设备，这种设备在一定年限内随着时间推移逐渐损坏。所以，保留这种设备的时间越长，每年的维修费用就越大。现我们假设该公司在第一年开始时必须购置一台这种设备，并假设计划使用这台设备的时间为五年，估计这台设备的购买费和维修费(单位：万元)见表 12-1。

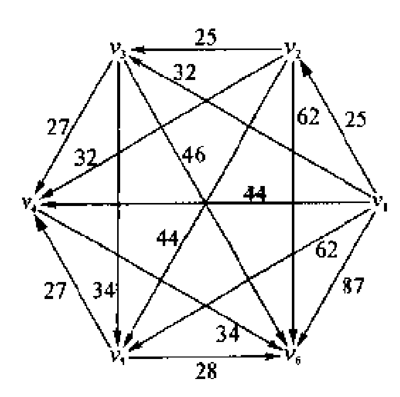
表 12-1(a)　第一年到第五年的购买价格

年　号	1	2	3	4	5
价　格	20	20	22	22	23

表 12-1(b)　不同使用年限的设备的维修费

使用年限	0—1	1 2	2—3	3—4	4－5
维　修　费	5	7	12	18	25

这家公司希望确定应在哪一年购买一台新设备，使得维修费和新设备的购置费的总和最小。



新 12-3　设备更新问题

解：考虑六个点 v_1，v_2，v_3，v_4，v_5，v_6，其中 $v_i(i=1,\cdots,5)$ 表示在第 i 年初要购买新设备。v_6 是虚设点，表示在第 5 年底才购买新设备。再从点 $v_i(i=1,\cdots,5)$ 引出指向点 v_{i+1}，v_{i+2}，…，v_6 的弧，弧(v_i, v_j)表示第 i 年年初购进的新设备要使用到第 j 年 $(j=2,\cdots,6)$ 的年初。弧 (v_i, v_j)上所赋的权为第 i 年的购置费加上从第 i 年初到第 j 年初这段时间的维修总费用。比如 $W(1,4)=20+(5+7+12)=44$(万元)，如此计算可得到所有权值，见下面的赋权有向图(图 12-3)。

本问题变为在上面的赋权图中求一条从 v_1 到 v_6 总权最小的路径。

求解如下

```
》clear; n=6; w=inf*ones(6);
》w(1,[2,3,4,5,6])=[25,32,44,62,87];
》w(2,[3,4,5,6])=[25,32,44,62];
》w(3,[4,5,6])=[27,34,46]; w(4,[5,6])=[27,34];
》w(5,6)=28; [s,d]=minroute(1,n,w)
```

```
s=        %输出,每列表示最短路径的顶点序号
      1      1      1      1      1      1
      0      2      3      4      5      3
      0      0      0      0      0      6
d=        %最短路径的权值
      0      25     32     44     62     78
```

可见从 v_1 到 v_6 总权最小的路径为 $v_1 \to v_3 \to v_6$，权值为 78。由图 12-3 可以看出 $v_1 \to v_4 \to v_6$ 也是一条总权最小的路径权值为 78，可知最小路径不唯一。

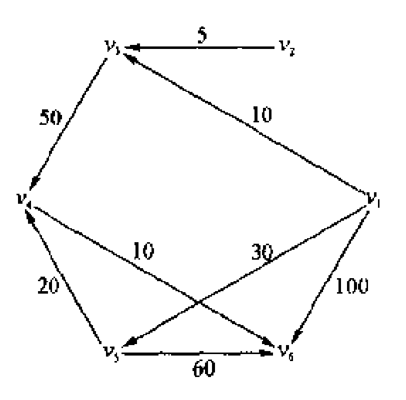


图 12-4 例 3 图

例 3 求图 12-4 中，自点 v_1 到其他各点的最短有向路。

解：运用 Dijkstra 算法程序计算如下

```
》clear; w = inf* ones(6); w(1, 3) = 10; w(1, 5) = 30;
》w(1, 6) = 100; w(2, 3) = 5; w(3, 4) = 50; w(4, 6) = 10;
》w(5, 4) = 20; w(5, 6) = 60;
》i = 1; [s, d] = minroute(i, 6, w)
s=        %输出,每列表示最短路径的顶点序号
      1      1      1      1      1      1
      0      3      5      5      5      2
      0      0      0      4      4      0
      0      0      0      0      6      0
d=        %最短路径的权值
      0      10     30     50     60     Inf
```

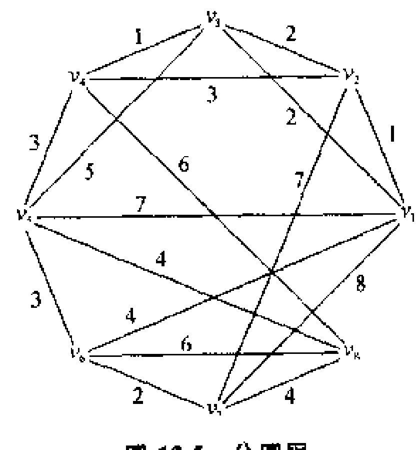


图 12-5 公路网

例 4 8 个城市之间有公路网，每条公路为图 12-5 中的边，边上的权数表示通过该公路所需的时间。设你处在城市 v_1，那么从该城市到其他各城市，应选择什么路径使所需的时间最少？

解：这是一个无向网，根据问题题意是要求一条从 v_1 到其他各城市的最短路径，求解如下

```
》clear; w = inf* ones(8);
》w(1, [2, 3, 5, 6, 7]) = [1, 2, 7, 4, 8];
》w(2, [1, 3, 4, 7]) = [1, 2, 3, 7];
》w(3, [1, 2, 4, 5]) = [2, 2, 1, 5];
w(4, [2, 3, 5, 8]) = [3, 1, 3, 6];
》w(5, [1, 3, 4, 6, 8]) = [7, 5, 3, 3, 4];
》w(6, [1, 5, 7, 8]) = [4, 3, 2, 6]; w(7, [1, 2, 6, 8]) = [8, 7, 2, 4];
》w(8, [4, 5, 6, 7]) = [6, 4, 6, 4];
》[s, d] = minroute(1, 8, w)
s=        %输出
```

1　1　1　1　1　1　1　1

0　2　3　3　6　3　6　3

0　0　0　4　0　4　7　4

0　0　0　0　0　5　0　8

d=

0　1　2　3　4　6　6　9

由 s 可见从 v_1 到其他各城市的最短路径，d 为相应的权值。

§12.5　实 验 习 题

1　某企业使用一种设备，每年年初，决定是购置新的，还是继续用旧的。若购置新设备，则需付一定的购置费用；若继续使用旧设备，则需付一定的维修费用。假设该企业在第一年开始时必须购置一台这种设备，并假设计划使用这种设备的时间为五年。表 12-2 给出一台新设备的价格以及一台设备的使用维修费用。

表 12-2(a)　从第 1 年到第 5 年的设备价格(单位:千元)

年　　号	1	2	3	4	5
价　　格	11	11	12	12	13

表 12-2(b)　不同使用年限的设备的维修费(单位:千元)

使用年限	0—1	1—2	2—3	3—4	4—5
维　修　费	5	6	8	11	18

这家企业希望确定应在哪一年购买一台新设备，使得维修费和新设备的购置费的总和最小。

2　求图 12-6 中 a 到各顶点的最短距离和最短路径。

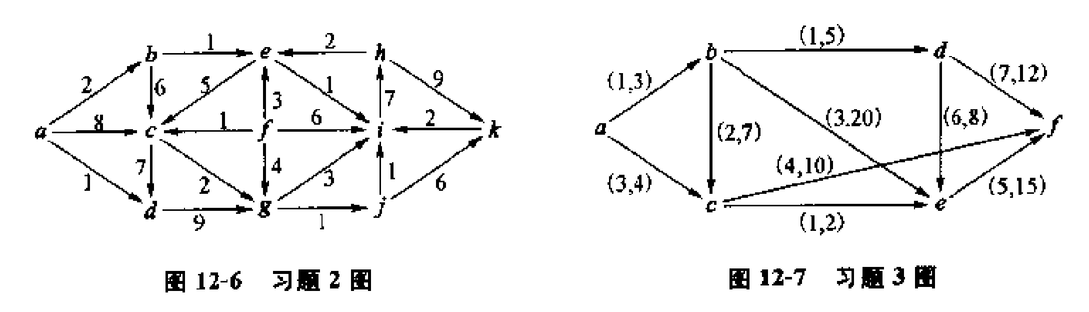


图 12-6　习题 2 图　　　图 12-7　习题 3 图

3　一辆货车从水泥厂运水泥至某建筑工地。如图 12-7，图中 a 表示水泥厂所在处，f 为建筑工地所在处，图中弧旁括弧内的数字，第一个数表示两点间距离，第二个数表示两点间汽车行驶所需时间。试分别依据最短距离和最少行驶时间确定水泥厂至某建筑工地的汽车行驶路线?

4　已知 8 口海上油井，相互间距离如表 12-3。已知 1 号井离海岸最近，为 5 海里。问从海岸经 1 号井铺设油管将各油井连接起来，应如何铺设使输油管长度为最短(为便于计量

和检修,油管只准在各井位处分叉)。

表 12-3 各油井间距离(单位:海里)

	2	3	4	5	6	7	8
1	1.3	2.1	0.9	0.7	1.8	2.0	1.5
2		0.9	1.8	1.2	2.6	2.3	1.1
3			2.6	1.7	2.5	1.9	1.0
4				0.7	1.6	1.5	0.9
5					0.9	1.1	0.8
6						0.6	1.0
7							0.5

实验十三　生产计划的制定　动态规划

本实验的目的通过将应用问题看作多阶段决策过程来建立数学模型，复习和进一步了解动态规划的相关知识及其求解的逆序算法，并结合计算机编程(特别是运用 MATLAB)解决实际问题。强化动态规划应用从建模到求解的全过程实验活动。本实验研究了动态规划的几个典型用法，如生产计划制定、确定最短路径和设备分配等问题，并给出了动态规划逆序算法的 MATLAB 参考程序。

§13.1　引例：生产计划的制定

某工厂与用户订立合同，在四个月内出售一定数量的某种产品，产量限制为 10 的倍数，工厂每月最多生产 100 件，产品可以存储，存储费用为每台 2 百元，每个月的需求量及每件产品的生产成本见表 13-1。

表 13-1　生产成本和需要量

月　份	每件生产成本(百元)	需要量(件)
1	70	60
2	72	70
3	80	120
4	76	60

现在分别在①1 月初没有存货可用和②1 月初有 20 件存货可用这两种情况下确定每月的生产量，要求既能满足每月的合同需求量，又使生产成本和存储费用达到最小。

静态地看，本引例是一个整数规划问题，但这里我们可以把这个问题的解决动态地视为各月(一般称为**阶段**)先后作出**决策**(这里指生产量)的过程——**多阶段的决策过程**，而在每个月作决策时，不能仅考虑本月的费用(一般称为**阶段指标**)，因为本月的决策会对以后各月的决策产生影响，因此应考虑从本月直到第四月末的总费用(**总指标**)，而每月的决策依赖于各月初仓库中的存货量(一般称为**始端**)而和以前各月如何造成这存货量的情况无关(称为**无后效性**)。在例 2 中我们将计算当 1 月初无存货可用时的**最优决策**见表 13-2。

表 13-2　最优生产计划

	1　月	2　月	3　月	4　月
月初存储量(件)	0	40	70	0
产量(件)	100	100	50	60

则第四月的决策即为月初仓储数为 0 时的最优决策，第三、四月的决策即为第三月初仓储数为 70 时的最优决策，以及第二、三、四月的决策即为第二月初仓储数为 40 时的最优决策。

因为若不然，如对应于第三月初仓储数为 70 时，第三、四月的最优决策是分别生产 80 件和 30 件（即这样的费用比分别生产 50 件和 60 件更省），则我们保留第一、二月的生产数，而把第三、第四月分别改为 80 件和 30 件，这个方案显然优于原来的方案，这和原来的方案是最优相矛盾。这个性质可以简述为：**最优决策的任何截断仍是最优的（最优性原理）**。把这一最优化问题视为符合最优性原理、无后效性的多阶段决策过程并进行求解的方法称为**动态规划方法**。

§13.2 动态规划的态本理论复习

1. 基本思想与逆序解法的直观回顾

前而我们已简单介绍了动态规划。为了更便了了解动态规划的基本思想、描述方式和逆序解法，我们来看一个确定网络最短路径问题的例子。

例 1 （*最短路径的确定*）以下是一个赋权图（网络），两顶点连线上的数字表示距离，确定一条从始点 v_1 到终点 v_{16} 铺设管道并使总距离最小的路线（最短路径）。如图 13-1 所示。

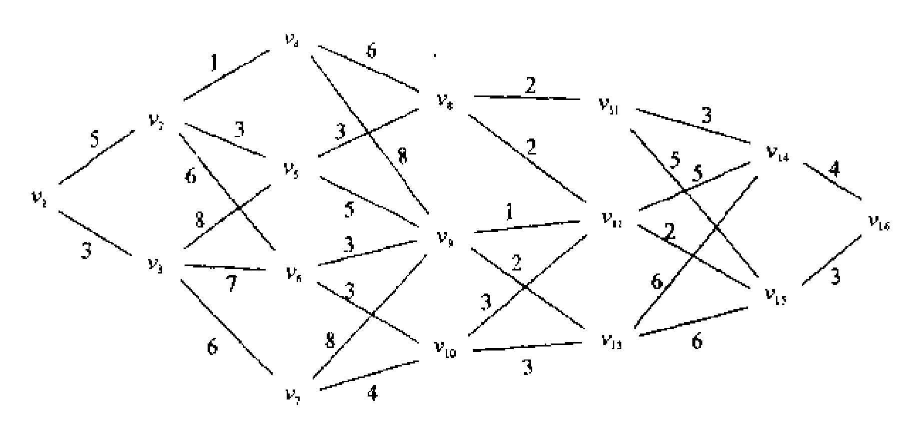


图 13-1 例 1 图

直观上我们有这样一个重要常识：如果由起点 v_1 经过 v_i 点和 v_j 点而到达终点 v_{16} 是一条最短路线，则由点 v_i 出发经过 v_j 点到达终点 v_{16} 的这条路线，对于从 v_i 点出发到达终点的所有可能选择的不同路线来说，必定也是最短路线。例如在最短路线问题中，若找到了 $v_1 \to v_2 \to v_5 \to v_8 \to v_{12} \to v_{15} \to v_{16}$ 是由 v_1 点到 v_{16} 点最短路线，则 $v_8 \to v_{12} \to v_{15} \to v_{16}$ 应该是由 v_8 点出发到达终点的所有可能选择的不同路线中最短路线。这一特征即为上而所提到的**最优性原理：最优决策的任何截断仍是最优的**，这是动态规划的基本原理。

根据最优性原理，寻找最短路线可从最后一段开始，用由后向前逐步递推的方法，求出各点到 v_{16} 点的最短路线，最后求得由 v_1 点到 v_{16} 点的最短路线，所以动态规划的**逆序求解方法**是从终点逐段向始点方向寻找最短路线的一种方法。下面逐段完成计算。

例 1 当然可以用图与网络实验中介绍的 Dijkstra 算法来求解，但这里我们将把该问题看成 6 个阶段的决策过程，并逆序逐段求解，从而较直观地揭示动态规划的基本思想。

$k=6$ 时，以 $f_6(v_{14})$ 表示由 v_{14} 到 v_{16} 的最短距离，以 $f_6(v_{15})$ 表示由 v_{15} 到 v_{16} 的最短距离，则 $f_6(v_{14})=4$，$f_6(v_{15})=3$。

$k=5$ 时，出发点有三个 v_{11}、v_{12} 和 v_{13}。若从 v_{11} 出发有 v_{14} 和 v_{15} 两个选择，以 $f_5(v_{11})$ 表示由 v_{11} 到 v_{16} 的最短距离，$d_5(v_{11}, v_{14})$ 表示由 v_{11} 到 v_{14} 的距离，$d_5(v_{11}, v_{15})$ 表示由 v_{11} 到 v_{15}

的距离，$u_5(v_{11})$表示相应的选择或决策，则：

$$f_5(v_{11}) = \min\{d_5(v_{11}, v_{14}) + f_6(v_{14}), d_5(v_{11}, v_{15}) + f_6(v_{15})\}$$
$$= \min\{3+4, 5+3\} = 7$$

可见 $u_5(v_{11}) = v_{14}$，其最短路径为 $v_{11} \to v_{14} \to v_{16}$。

同理，从 v_{12} 出发也有 v_{14} 和 v_{15} 两个选择，$f_5(v_{12})$、$d_5(v_{12}, v_{14})$、$d_5(v_{11}, v_{15})$和 $u_5(v_{12})$ 意义与上面相似，则：

$$f_5(v_{12}) = \min\{d_5(v_{12}, v_{14}) + f_6(v_{14}), d_5(v_{12}, v_{15}) + f_6(v_{15})\}$$
$$= \min\{5+4, 2+3\} = 5$$

可见 $u_5(v_{12}) = v_{15}$，其最短路径为 $v_{12} \to v_{15} \to v_{16}$。

从 v_{13} 出发，同样有：

$$f_5(v_{13}) = \min\{d_5(v_{13}, v_{14}) - f_6(v_{14}), d_5(v_{13}, v_{15}) + f_6(v_{15})\}$$
$$= \min\{6+4, 6+3\} = 9$$

可见 $u_5(v_{13}) = v_{15}$，其最短路径为 $v_{13} \to v_{15} \to v_{16}$。

$k = 4$ 时，有三个出发点 v_8、v_9 和 v_{10}，同样计算如下：

$$f_4(v_8) = \min\{d_4(v_8, v_{11}) + f_5(v_{11}), d_4(v_8, v_{12}) + f_5(v_{12})\}$$
$$= \min\{2+7, 2+5\} = 7$$

可见 $u_4(v_8) = v_{12}$，其最短路径为 $v_8 \to v_{12} \to v_{15} \to v_{15}$。

$$f_4(v_9) = \min\{d_4(v_9, v_{12}) + f_5(v_{12}), d_4(v_9, v_{13}) + f_5(v_{13})\}$$
$$= \min\{1+5, 2+9\} = 6$$

可见 $u_4(v_9) = v_{12}$，其最短路径为 $v_9 \to v_{12} \to v_{15} \to v_{16}$。

$$f_4(v_{10}) = \min\{d_5(v_{10}, v_{12}) + f_5(v_{12}), d_4(v_{10}, v_{13}) + f_5(v_{13})\}$$
$$= \min\{3+5, 3+9\} = 8$$

可见 $u_4(v_{10}) = v_{12}$，其最短路径为 $v_{10} \to v_{12} \to v_{15} \to v_{16}$。

$k = 3$ 时，同样计算有：

$$f_3(v_4) = 13, u_3(v_4) = v_8; f_3(v_5) = 10, u_3(v_5) = v_8$$

$$f_3(v_6) = 9, u_3(v_6) = v_9; f_3(v_7) = 12, u_3(v_7) = v_{10}$$

$k = 2$ 时，同样计算有：

$$f_2(v_2) = 13, u_2(v_2) = v_8; f_2(v_3) = 16, u_2(v_3) = v_6$$

$k = 1$ 时，只有一个出发点 v_1，则：

$$f_1(v_1) = \min\{d_1(v_1, v_2) + f_2(v_2), d_1(v_1, v_3) + f_2(v_3)\}$$
$$= \min\{5+13, 3+16\} = 18$$

可见 $u_1(v_1) = v_2$，所以本题的最短路径为 $v_1 \to v_2 \to v_5 \to v_8 \to v_{12} \to v_{15} \to v_{16}$。

2. 动态规划的基本概念及其数学描述

(1) 阶段　整个问题的解决可分为若干个相互联系的阶段依次进行。通常按时间或空间划分阶段，描述阶段的变量称为**阶段变量**，记为 k。

(2) 状态　状态表示每个阶段开始所处的自然状况或客观条件，它描述了研究问题过程的状况。各阶段的状态通常用**状态变量描述**。常用 x_k 表示第 k 阶段的状态变量。n 个阶段的决策过程有 $n+1$ 个状态。用动态规划方法解决多阶段决策问题时，要求整个过程具有**无后效性**，即：如果某阶段的状态给定，则此阶段以后过程的发展不受以前状态的影响，未来状态只依赖于当前状态。

(3) 决策　某一阶段的状态确定后，可以作出各种选择从而演变到下一阶段某一状态，这种选择手段称为**决策**。描述决策的变量称为**决策变量**。决策变量限制的取值范围称为**允许决策集合**。用 $u_k(x_k)$ 表示第 k 阶段处于状态 x_k 时的决策变量，它是 x_k 的函数，用 $D_k(x_k)$ 表示 x_k 的允许决策的集合。比如在例 1 中，$D_2(x_2)=\{v_4, v_5, v_6\}$。

(4) 策略　一个由每个阶段的决策按顺序排列组成的集合称为**策略**，用 p 表示。即 $p(x_1)=\{u_1(x_1), u_2(x_2), \cdots, u_n(x_n)\}$。由第 k 阶段的状态 x_k 开始到终止状态的后部子过程的策略记为 $p_k(x_k)$，即 $p_k(x_k)=\{u_k(x_k), u_{k+1}(x_{k+1}), \cdots, u_n(x_n)\}$。在实际问题中，可供选择的策略有一定范围，此范围称为**允许策略集合**。允许策略集合中达到最优效果的策略称为**最优策略**。

(5) 状态转移方程　如果第 k 个阶段状态变量为 x_k，作出的决策 u_k，那么第 $k+1$ 阶段的状态变量 x_{k+1} 也被完全确定。用状态转移方程表示这种演变规律，写作

$$x_{k+1}=T_k(x_k, u_k)。$$

(6) 指标函数和最优值函数　**指标函数**是系统执行某一策略所产生结果的数量表示，是用来衡量策略优劣的数量指标，它定义在全过程和所有后部子过程上，分别用 V 和 V_k 表示。即

$$V(u_1, u_2, \cdots, u_n, x_1, x_2, \cdots, x_{n+1}) \text{和} V_k(u_k, \cdots, u_n, x_k, \cdots, x_{n+1})。$$

过程在某阶段 j 的**阶段指标函数**(或**阶段效益**)是衡量该阶段决策优劣的数量指标，它取决于状态 x_j 和决策 u_j，用 $v_j(u_j, x_j)$ 表示。如例 1 中两顶点间的距离是阶段指标函数，而指标函数则是直到终点 v_{16} 的距离的和。根据不同的实际问题，效益可以是利润、距离、产量或资源等。指标函数往往是各阶段效益的某种形式。指标函数的最优值称为**最优函数**。

(7) 最优策略和最优轨线　使指标函数 V_k 达到最优值的策略是从阶段 k 开始的后部子过程的最优策略，记为 $p_k^*=\{u_k^*, \cdots, u_n^*\}$。$p^*$ 是全过程的最优策略，简称为**最优策略**。从初始状态 $x_1(=x_1^*)$ 出发，过程按照 p^* 和状态转移方程演变所经历的状态序列 $\{x_1^*, \cdots, x_{n+1}^*\}$ 称为**最优轨线**。

§13.3　动态规划逆序算法的 MATLAB 程序

1. 逆序算法的基本方程

由例 1 的求解过程可以看出下面的方程在动态规划逆序求解中起着本质的作用

$$\begin{cases} f_k(x_k) = \min\{v_k(x_k, u_k(x_k)) + f_{k+1}(x_{k+1}) \mid u_k \in D_k(x_k)\} \\ f_{n+1}(x_{n+1}) = 0,\ x_{k-1} = T_k(x_k,\ u_k),\ k = n,\ n-1,\ \cdots,\ 1 \end{cases}$$

称此为动态规划逆序求解的**基本方程**。

可以把建立动态规划模型归纳成以下几个步骤

(1) 将问题恰当地划分为若干个阶段；

(2) 正确选择状态变量，使它既能描述过程的演变，又满足无后效性；

(3) 规定决策变量，确定每个阶段的允许决策集合；

(4) 写出状态转移方程；

(5) 确定各阶段各种决策的阶段指标，列出计算各阶段最优后部策略指标的基本方程。

2. 动态规划逆序算法的 MATLAB 程序 DynProg. m

```
                    %M 函数 dynprog.m
function  [p_opt, fval] = dynprog(x, DecisFun, ObjFun, TransFun)
%  [p_opt, fval] = dynprog(x, DecisFun, ObjFun, TransFun)
%  自由始端和终端的动态规划,求指标函数最小值的逆序算法递归
%  计算程序。x是状态变量,一列代表一个阶段状态;M函数
%  DecisFun(k, x)由阶段k的状态变量x求出相应的允许决策变量;
%  M函数ObjFun(k, x, u)是阶段指标函数,M函数TransFun(k, x, u)
%  是状态转移函数,其中x是阶段k的某状态变量,u是相应的决策变量;
%  输出p_opt由4列构成,p_opt=[序号组;最优轨线组;最优策略组;
%  指标函数值组];fval是一个列向量,各元素分别表示p_opt各
%  最优策略组对应始端状态x的最优函数值;
k = length(x(1, :)); f_opt = nan* ones(size(x)); d_opt = f_opt;
t_vubm = inf* ones(size(x)); x_isnan = ~isnan(x); t_vub = inf;
%  计算终端相关值
tmp1 = find(x_isnan(:, k)); tmp2 = length(tmp1);
for i = 1:tmp2
    u = feval(DecisFun, k, x(i, k)); tmp3 = length(u);
    for j = 1:tmp3
        tmp = feval(ObjFun, k, x(tmp1(i), k), u(j));
        if tmp <= t_vub,
          f_opt(i, k) = tmp; d_opt(i, k) = u(j); t_vub = tmp;
end; end; end
%  逆推计算各阶段的递归调用程序
for ii = k-1: -1:1
    tmp10 = find(x_isnan(:,ii)); tmp20 = length(tmp10);
    for i = 1:tmp20
        u = feval(DecisFun, ii, x(i, ii)); tmp30 = length(u);
        for j = 1:tmp30
```

```
            tmp00 = feval(ObjFun, ii, x(tmp10(i), ii), u(j));
            tmp40 = feval(TransFun, ii, x(tmp10(i), ii), u(j));
            tmp50 = x(:, ii + 1) - tmp40;
            tmp60 = find(tmp50 == 0);
            if ~isempty(tmp60),
              tmp00 = tmp00 + f_opt(tmp60(1), ii + 1);
              if tmp00 <= t_vubm(i, ii)
                  f_opt(i, ii) = tmp00; d_opt(i, ii) = u(j);
                  t_vubm(i, ii) = tmp00;
end; end; end; end; end;
fval = f_opt(tmp1, 1);
%  记录最优决策、最优轨线和相应指标函数值
p_opt = []; tmpx = []; tmpd = []; tmpf = [];
tmp0 = find(x_isnan(:, 1));tmp01 = length(tmp0);
for i = 1:tmp01,
    tmpd(i) = d_opt(tmp0(i), 1);
    tmpx(i) = x(tmp0(i), 1);
    tmpf(i) = feval(ObjFun, 1, tmpx(i), tmpd(i));
    p_opt(k*(i-1)+1, [1, 2, 3, 4]) = [1, tmpx(i), ...
tmpd(i), tmpf(i)];
    for ii = 2:k
        tmpx(i) = feval(TransFun, ii-1, tmpx(i), tmpd(i));
        tmp1 = x(:, ii) - tmpx(i); tmp2 = find(tmp1 == 0);
        if ~isempty(tmp2)
            tmpd(i) = d_opt(tmp2(1), ii);
        end;
        tmpf(i) = feval(ObjFun, ii, tmpx(i), tmpd(i));
        p_opt(k*(i-1) + ii, [1, 2, 3, 4]) = [ii, tmpx(i), ...
tmpd(i), tmpf(i)];
end; end;
```

§13.4 实 验 例 题

下面将就动态规划的几个典型应用问题分别举例计算。

例 2 (生产计划制定)

这是一个 4 阶段动态规划问题。如果用逆序法解题,第 1 阶段是 1 月份,…,第 4 阶段是 4 月份。

x_k——第 k 阶段开始的产品存储数(状态变量);

u_k——第 k 阶段的产量(决策变量);

c_k——第 k 阶段每件产品的生产成本；

q_k——第 k 阶段的需求量；

阶段指标的函数为：$v_k(x_k, u_k) = c_k u_k + 2x_k$；

状态转移方程为：$x_{k+1} = x_k + u_k - q_k$；

基本方程为：

$$\begin{cases} f_4(x_4, u_4) = v_4(x_4, u_4) \\ f_k(x_k, u_k) = \min\{v_k(x_k, u_k) + f_{k+1}(x_{k+1}) \mid u_k \in D_k(x_k)\}, \ k = 3, 2, 1。\end{cases}$$

对于本例的问题①：一月初无存货，可首先分析出各月的最大存储量和产量。如对 4 月初，前三个月的最大产量为 300 件(每月最大产量为 100 件)，实际需求量为 60＋70＋120＝250(件)，所以四月初的最大存储量为 300－250＝50 件。因产量限制为 10 的倍数，4 月初的存储量只可能是 0、10、20、30、40、50 这六种。而 4 月份的实际需要量为 60 件，因此第一阶段的产量(决策)相应为 60、50、40、30、20、10，进而计算 $f_4(x_4, u_4)$，再分析 3 月初情况等等，如此可仿例 1 逐段手工计算求出最优决策。

下面将调用参考程序 dynprog.m 进行计算。由于计算机的优势，这里将就 1 月初存货分别为 0、10、20、30、40、50、60 的所有可能情况进行计算。把此问题作为自由始端动态规划考虑，此时各阶段的最大存货出现在 $x_1 = 60$ 时，经分析可知 $0 \leqslant x_2 \leqslant 100$，$0 \leqslant x_3 \leqslant 130$，$0 \leqslant x_4 \leqslant 60$，考虑到产量是 10 的倍数，根据上面所述的阶段指标函数、状态转移方程和基本方程，写出下面的 3 个 M 函数以备计算时调用，函数意义见函数的说明部分。

```
    %M 函数 eg13f1_2.m
function u = DecisF_1(k, x)
%   在阶段 k 由状态变量 x 的值求出其相应的决策变量所有的取值
c = [70, 72, 80, 76]; q = 10*[6, 7, 12, 6];
if q(k) - x<0, u = 0:100;          %决策变量不能取为负值
else, u = q(k) - x:100; end;       %产量满足需求且不超过 100
u = u(:);
```

```
    %M 函数 eg13f2_2.m
function v = ObjF_1(k, x, u)
%   阶段 k 的指标函数
c = [70, 72, 80, 76]; v = c(k)*u + 2*x;
```

```
    %M 函数 eg13f3_2.m
function y = TransF_1(k, x, u)
%   状态转移方程
q = 10*[6, 7, 12, 6]; y = x + u - q(k);
```

调用 dynprog.m 计算如下

```
》clear; x = nan* ones(14, 4); % x 是 10 的倍数,最大范围 0 ≤ x ≤ 130,
        %  因此 x = 0, 1, …, 13 ,所以 x 初始化取 14 行,nan 表示无意义元素
》x(1:7, 1) = 10* (0:6)';   % 按月定义 x 的可能取值
》x(1:11, 2) = 10* (0:10)'; x(1:12, 3) = 10* (2:13)';
》x(1:7, 4) = 10* (0:6)';
》[p, f] = dynprog(x, 'eg13f1_2', 'eg13f2_2', 'eg13f3_2')
p =   %输出结果
        1        0      100     7000
        2       40      100     7280
        3       70       50     4140
        4        0       60     4560
        1       10      100     7020
        2       50      100     7300
        3       80       40     3360
        4        0       60     4560
        1       20      100     7040
        2       60      100     7320
        3       90       30     2580
        4        0       60     4560
        1       30      100     7060
        2       70      100     7340
        3      100       20     1800
        4        0       60     4560
        1       40      100     7080
        2       80      100     7360
        3      110       10     1020
        4        0       60     4560
        1       50      100     7100
        2       90      100     7380
        3      120        0      240
        4        0       60     4560
        1       60      100     7120
        2      100      100     7400
        3      130        0      260
        4       10       50     3820

f =
     22980
     22240
     21500
     20760
```

20020

19280

18600

由 p 的第一和第三个 4 行可以看出一月初无存货和有存货 20 件的最优决策，现将其列成下表 13-3，以供对比理解。

表 13-3

1 月初的存货量	阶段序号（月份）	最优轨线（储存量）	最优决策（产量）	阶段指标函数值（成本）
0（件）	1	0	100	7000
	2	40	100	7280
	3	70	50	4140
	4	0	60	4560
20（件）	1	20	100	7040
	2	60	100	7320
	3	90	30	2580
	4	0	60	4560

由 f 的第 1、3 行可以看出对应的 4 个月的总成本分别为 22980 和 21500。

例 3　调用 dynprog.m 计算例 1 中网络的最短路径。

解：首先编写 3 个 M 函数，以供计算时调用。由图 13-1 可知状态变量有 7 个，并可写出下面的函数。

```
    %M 函数 eg13f1_3.m
function u = eg13f1_3(k, x)
%   在阶段 k 由状态变量 x 的值求出其相应的决策变量所有的取值
%   这里求出了所有状态 x 对应的决策变量值 u
if x == 1, u = [2; 3]; elseif x == 2, u = [4; 5; 6];
elseif x == 3, u = [5; 6; 7]; elseif (x == 4)|(x == 5), u = [8; 9];
elseif (x == 6)|(x == 7), u = [9; 10]; elseif x == 8, u = [11; 12];
elseif (x == 9)|(x == 10), u = [12; 13];
elseif (x == 11)|(x == 12)|(x == 13), u = [14; 15];
elseif (x == 14)|(x == 15), u = 16; elseif x == 16, u = 16;
end

    %M 函数 eg13f2_3.m
function v = eg13f2_3(k, x, u)
%   各阶段指标函数值，如 x = 1, u = 2 时 v = 5。
tt = [5; 3; 1; 3; 6; 8; 7; 6; 6; 8; 3; 5; 3; 3; 8; 4; ...
      2; 2; 1; 2; 3; 3; 3; 5; 5; 2; 6; 6; 4; 3];
```

```
tmp = [x == 1&u == 2, x == 1&u == 3, x == 2&u == 4, x == 2&u == 5, ...
     x == 2&u == 6, x == 3&u == 5, x == 3&u == 6, x == 3&u == 7, ...
     x == 4&u == 8, x == 4&u == 9, x == 5&u == 8, x == 5&u == 9, ...
     x == 6&u == 9, x == 6&u == 10, x == 7&u == 9, x == 7&u == 10, ...
     x == 8&u == 11, x == 8&u == 12, x == 9&u == 12, x == 9&u == 13, ...
     x == 10&u == 12, x == 10&u == 13, x == 11&u == 14, x == 11&u == 15, ...
     x == 12&u == 14, x == 12&u == 15, x == 13&u == 14, x == 13&u == 15, ...
     x == 14&u == 16, x == 15&u == 16];
v = tmp* tt;
```

```
    %M 函数 eg13f3_3.m
function y = eg13f3_3(k, x, u)
y = u;                                                  % 状态转移方程
```

计算如下：

```
》clear; x = nan* ones(4, 7);                     % 初始化，nan 为无意义元素
                                                % 7 个状态变量，每个最多 4 种取值
》x(1, 1) = 1; x(1:2, 2) = [2; 3]; % 逐列定义 x,其值为顶点的序号
》x(1:4, 3) = (4:7)'; x(1:3, 4) = (8:10)'; x(1:3,5) = (11:13)';
》x(1:2,6) = [14; 15]; x(1, 7) = 16;
》[p, f] = dynprog(x, 'eg13f1_3', 'eg13f2_3', 'eg13f3_3')
p =    %输出结果
           1        1        2        5
           2        2        5        3
           3        5        8        3
           4        8       12        2
           5       12       15        2
           6       15       16        3
           7       16       16        0

f =
       18   %最短路径距离
```

可见最短路径按顶点序号为 1→2→5→8→12→15→16。

例 4　某工业部门根据国家计划的安排，拟将 5 台某种高效率的设备，分配给所属的甲、乙、丙三个工厂，各工厂若获得这种设备之后，可以为国家提供的盈利如下面的表13-4，问：这 5 台设备如何分配给各工厂才能使国家得到的盈利最大？

解：将问题按工厂分为三个阶段，甲、乙和丙 3 个工厂分别编号为 1、2 和 3。

设状态变量 x_k 表示分配给第 k 个工厂至第 n 个工厂的设备台数。

决策变量 u_k 表示分配给第 k 个工厂的设备台数。

则状态转移方程 $x_{k+1} = x_k - u_k$，x_{k+1} 为分配给第 $k+1$ 个工厂至第 n 个工厂的设备

台数。

阶段指标函数 $v_k(u_k)$表示 u_k 台设备分配到第 k 个工厂所获得的盈利值。

$f_k(x_k)$表示 x_k 台设备分配给第 k 个工厂至第 n 个工厂所获得的最大盈利值。则基本方程为

$$\begin{cases} f_k(x_k) = \max\{v_k(u_k) + f_{k+1}(x_{k+1}) \mid u_k\},\ k = 2,\ 1 \\ f_3(x_3) = v_3(u_3) \end{cases}$$

表 13-4

设 备 数	甲	乙	丙
0	0	0	0
1	3	5	4
2	7	10	6
3	9	11	11
4	12	11	12
5	13	11	12

利用计算机计算的优势,可根据表 13-4 将本问题视为自由始端,即初始状态 x = 0、1、2、3、4、5 的动态规划问题求解。

由此可以写出下面 3 个 M 函数,以供计算时调用。

```
%M 函数 eg13f1_4.m
function u = eg13f1_4(k, x)
%  在阶段 k 由状态变量 x 的值求出其相应的决策变量所有的取值
if k == 3, u = x;                          %第三阶段丙工厂得到所有剩余设备
else, u = 0:x; end

%M 函数 eg13f2_4.m
function v = eg13f2_4(k, x, u)
%  阶段 k 的指标函数
w = [0, 0, 0; 3, 5, 4; 7, 10, 6; 9, 11, 11; 12, 11, 12; 13, 11, 12];
w = -w;                                   %  标准化,即将基本方程中 max 转化为 min
v = ([0, 1, 2, 3, 4, 5] == u)*w(:, k);        %  对此题可定义 v=w(u+1, k);

%M 函数 eg13f3_4.m
function y = eg13f3_4(k, x, u)
%  状态转移方程
y = x - u;
```

计算如下

```
》clear;
》x = [0; 1; 2; 3; 4; 5]; x = [x, x, x];
》[p, f] = dynprog(x, 'eg13f1_4', 'eg13f2_4', 'eg13f3_4')
p =

     1     0     0     0
     2     0     0     0
     3     0     0     0
     1     1     0     0
     2     1     1    -5
     3     0     0     0
     1     2     0     0
     2     2     2   -10
     3     0     0     0
     1     3     0     0
     2     3     2   -10
     3     1     1    -4
     1     4     2    -7
     2     2     2   -10
     3     0     0     0
     1     5     2    -7
     2     3     2   -10
     3     1     1    -4
f =

     0
    -5
   -10
   -14
   -17
   -21
```

由 p 和 f 可见，有 5 台设备时，可各分配给甲、乙和丙工厂 2、2 和 1 台设备，获最大盈利 21；有 4 台设备时可分配给甲、乙和丙工厂 2、2 和 0 台设备，获最大盈利 17；如此也可知有 3、2、1、0 台设备时的最优分配方案。

§13.5 实 验 习 题

1 某公司打算向它的三个营业区增设 4 个销售点，从各区赚取的利润与增设的销售店个数有关，其数据见表 13-5。

试求各区应分配几个增设的销售店，才能使总利润最大？其值是多少？

表 13-5

销售点增加数	第一区利润（万元）	第二区利润（万元）	第三区利润（万元）	第四区利润（万元）
0	160	190	200	250
1	310	225	298	308
2	541	445	399	487
3	600	517	601	655
4	705	632	721	674

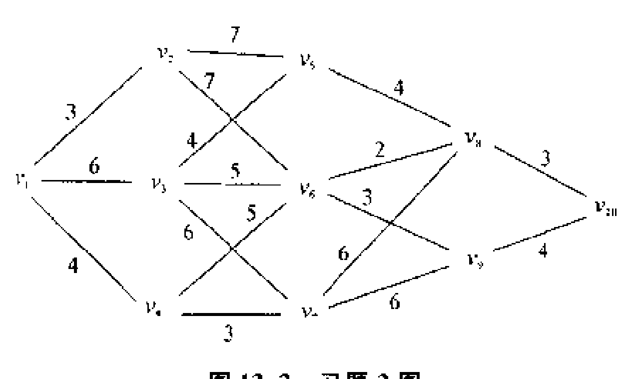


图 13-2 习题 2 图

2 图 13-2 为一线路网络，要铺设从 v_1 点到 v_{10} 点的电话线，中间需经过 3 个点。第 1 点可以是 v_2、v_3、v_4 中的某一个点，第 2 点可以是 v_5、v_6、v_7 中的一个点，第 3 点可以是 v_8、v_9 中的一个点。各点之间若能铺设电话线，则在图中以连线表示，连线旁的数字表示两点间的距离。用动态规划方法求一条从 v_1 至 v_{10} 的最短路线。

3 某船厂接受 7 艘船的订货，要求第一、二、三、四季度末分别交货 1、1、2、3 艘，该厂每季度的生产能力为 2 艘，预计各季度的单位生产成本为 4、6、7、9，各季度多生产的船可存在仓库中，其每季度每条船的保管费为 1.2，试制定一个完成订货合同且总费用最省的生产计划。

附录 A　MATLAB 函数命令索引表

函数名	含　　义	函数名	含　　义
abs	绝对值函数	bench	MATLAB 测试基准问题
acos	反余弦函数	blanks	设置一个由空格组成的字符串
acosh	反双曲余弦函数	bone	带有蓝色的灰度颜色表
acot	反余切函数	break	中断循环执行的语句
acoth	反双曲余切函数	brighten	使图形色调变亮
acsc	反余割函数	bucky	Buckminister Fuller 拱形演示
acsch	反双曲余割函数	caxis	伪颜色坐标轴设定
airfoil	NASA 翼面稀疏矩阵显示	cd	改变当前的工作目录
all	测试向量中所有元素是否为真	cdf2rd	复块对角矩阵到实块对角阵转换
angle	相角函数	cedit	设置命令行编辑与回调的参数
any	测试向量中是否有为真元素	ceil	对+∞方向取整数
ans	返回最新结果	census	2000 年美国人口普查预测
arith	MATLAB 的各种算术运算符信息	chol	Cholesky 分解
asec	反正割函数	cla	清除当前坐标轴
asech	反双曲正割	clabel	等高线剖面标志
asin	反正弦函数	clc	清除命令窗口显示
asinh	反双曲正弦	clear	删除内存中的变量与函数
atan	反正切函数	clf	清除当前图形窗口
atan2	四个象限内反正切	clock	时钟
atanh	反双曲正切	close	关闭图形窗口
auread	读声音文件	clommd	最小列的阶次
auwrite	写声音文件	colon	冒号表达式的帮助信息
axes	坐标轴任意形式的设定	colormap	设定颜色可查表
axis	坐标轴标度设定	colormenu	颜色表演示
balance	改进特征精度的均衡变换	colperm	由非零数据的计数来排列各列
bar	绘制条形图	comet	彗星状轨迹绘制

（续表）

函数名	含　　义	函数名	含　　义
comet3	绘制三维彗星状的轨迹	dbdown	改变局部工作空间内容
compan	生成伴随多项式矩阵	dbquit	退出跟踪调试模式
compass	绘制平面向量图	dbstack	列出函数调用关系
computer	计算机类型测试	dbstatus	列出所有的断点情况
cond	求矩阵的条件数	dbstep	跟踪调试单步执行
condest	估算条件数	dbstop	设置跟踪调试断点
conj	共轭复数函数	dbtype	列出带有命令行标号的 M 文件
contour	等高线图形绘制	dbup	改变局部工作空间内容
contour3	三维等高线绘制	deblank	消除字符串中的空格
contourc	等高线绘图计算	dec2hex	十进制到十六进制的转换
contrast	灰度对比度设置	deconv	因式分解与多项式除法
conv	求多项式乘法的卷积	del2	五点式离散 Laplace 变换
conv2	二维卷积	delete	删除文件
cool	天蓝粉色基色颜色表	delsqdemo	各种域上的有限差分演示
copper	线性铜色调颜色表	demo	运行 MATLAB 演示程序
corrcoef	相关函数系数	det	求矩阵的行列式
cos	余弦函数	diag	建立对角矩阵或获取对角向置
cosh	双曲余弦	diary	将 MATLAB 运行的命令存盘
cot	余切函数	diff	差分函数与近似微分
coth	双曲余切	diffuse	图象柔焦处理
cov	协方差矩阵	dir	列出当前目录的内容
cplxdemo	复变量函数映射函数演示	disp	显示矩阵成文本
cplxpair	将数据按共轭复数对重新排序	dmperm	Dulmage-Mendelsohn 分解
cputime	所用的 CPU 时时间	drawnow	刷新图象
csc	余割函数	earthmap	地球拓扑图形的显示
csch	双曲余割函数	echo	显示文件中的 MATLAB 命令
cumsum	各元素累加积	eig	求矩阵的特征值与特征向量
cylinder	产生柱体	eigmovie	对称矩阵特征值求解过程演示
date	日期	else	与 if 一起使用的转移语句
dbclear	清除跟踪调试断点	elseif	与 If 一起使用的转移语句
dbcont	跟踪调试恢复执行	end	结束控制语句块的命令

(续表)

函数名	含　　义	函数名	含　　义
eps	浮点相对差限	findstr	字符串查找
error	显示错误信息并中断函数	finite	若参数为有限元素则为真
errorbar	误差条型图绘制	fitdemo	非线性最优化拟合演示
etree	矩阵消元树结构	fix	对零方向取整数
etreeplot	绘制消元路径	flag	红白蓝黑基色颜色表
etime	所用时间的函数	fliplr	按左右方向翻转元素
eval	执行 MATLAB 语句构成的字符串	flipud	按上下方向翻转矩阵元素
exist	检验变量或文件是否已经定义	floor	对负无穷方向取整数
expm	矩阵指数函数	flops	浮点运算计数器
expm1	expm 函数的 M 文件实现	fmin	单变量函数最优化
expm2	Taylor 级数求矩阵指数	fmins	多变量函数最优化
expm3	特征值特征向量法求矩阵指数	fopen	打开文件
exp	指数函数	for	循环语句
eye	产生单位阵	format	设置输出格式
fclose	关闭文件	fourier	Fourier 级数展开图形演示
feather	羽状图形绘制	fplot	给定函数绘图
feof	测试文件是否结束	fplotdemo	函数图形绘制演示
ferror	查询文件输入输出错误状态	fprintf	有格式地向文件写入数据，参见 C
feval	执行字符串指定的文件	fread	从文件读入二进制数据
fft	离散 Fourier 变换	frewind	将文件指针至文件开头
fft2	二维离散 Fourier 变换	fscanf	从文件有格式地读入数据，参见 C
fftdemo	快速 Fourier 变换演示	fseek	设置文件位置指针
fftshift	取消谱中心零位	ftell	获得文件位置指针
fgetl	从文件读入一行数据(忽略换行)	full	由稀疏矩阵变换常规矩阵
fgets	从文件读入一行数据(保留换行)	function	MATLAB 函数表达式的引导符
figure	生成绘图窗口	funm	矩阵的任意函数
fill	绘制充填的二维多边形	fwrite	将二进制数据写入文件
fill3	绘制充填的三维多边形	fzero	单变量函数求根
filter	一维数字滤波	gallery	生成一些小的测试矩阵
filter2	二维数字滤波	gca	获得当前坐标轴的句柄
find	查找非零元素的下标	gcf	获得当前图形的窗口句柄

(续表)

函数名	含　　义	函数名	含　　义
get	获得对象属性	image	创建图象
getenv	获得环境参数	imagedemo	MATLAB 4.0 版图形处理功能演示
getframe	获得一幅"电影"图象	inf	无穷大(保留变量)
ginput	由鼠标器输入数据	info	显示 MATLAB 与 MathWorks 信息
global	定义全局变量	input	带有提示的键盘输入函数
gplot	绘制图论图形	int2str	整数转换为字符串
gray	线性灰度颜色表	interp1	一维插值(一维查表)
graymon	将图形窗口设置成灰度默认值	interp2	二维插值(二维查表)
grid	给图形加网格线或去除网格线	interpft	利用 FFT 的一维插值
griddata	插值用数据网格生成	intro	MATLAB 引言信息
gradient	近似梯度计算	inv	矩阵求逆
gtext	在鼠标指定的位置加文字说明	invhilb	生成逆 Hilbert 矩阵
hadamard	生成 Hadamard 矩阵	isempty	若参数为空矩阵,则结果为真
hankel	生成 Hankel 矩阵	isglobal	若参数为全局变量则为真
help	启动联机帮助文件显示	ishold	若屏幕处于保护状态则为真
hess	求取 Hessenberg 标准型	isieee	若有 IEEE 算术标准则为真
hex2num	十六进制到 IEEE 浮点数的转换	isinf	若参数为 Inf,则结果为真
hex2dec	十六进制到十进制的转换	isletter	若字符串为字母组成则为真
hidden	网格图隐含线设置开关	isnan	若参数为 NaN,则结果为真
hilb	生成 Hilbert 矩阵	issparse	若矩阵为稀疏表示则为真
hist	直方图绘制	isstr	若参数为字符串,则结果为真
hold	当前图形保护模式	jet	HSV 色调的变化型
home	将光标移动到左上角位置	keyboard	启动键盘管理程序
hostid	MATLAB 版务器的主机代号	knot	围绕三维结的柱形显示
hot	黑红黄白基色颜色表	kron	Kronecker 乘积函数
hsv	色度饱和值(HSV)颜色表	lasterr	查询上一条错误信息
hsv2rgb	HSV 对 RGB 颜色的转换	length	查询向量的维数
if	条件转移语句	life	Conway 生命假设的 MATLAB 版
ifft	离散 Fourier 逆变换	linspace	构造等距分布的向量
ifft2	二维离散 Fourier 逆变换	load	从文件中读入变量
imag	求取虚部函数	log	自然对数函数

（续表）

函数名	含　　义	函数名	含　　义
log10	常用对数函数	nnls	非零最小二乘
loglog	全对数坐标图绘制	nnz	非零元素个数
logm	矩阵的对数	nonzeros	非零元素
logspace	构造等对数分布的向量	norm	求矩阵的范数
lookfor	对 HELP 信息中的关键词查找	normest	估算范数
lorenz	Lorenz 混沌吸引子的曲线	null	右零空间
lower	将一个字符串内容转换为小写	num2str	将数值转换为字符串
lscov	最小二乘方差	nzmax	允许的非零元素存储空间
lu	矩阵的三角(LU 分解)	ode23	微分方程低阶数值解法
magic	生成魔术矩阵	ode23p	微分方程低阶数值解法并画图
matlabrc	启动主程序	ode45	数分方程高阶数值解法
max	求向量中最大元素	odedemo	常微分方程演示
mean	求向量各元素均值	ones	产生元素全部为 1 的矩阵
median	求向量各元素中间值	orient	设置打印纸方向
membrane	产生 Math Works 公司标志	orth	正交空间
menu	产生用户输入的菜单	pack	整理工作空间内存
mesh	三维网格图形	patch	低级填充多边形绘制函数
meshc	带有等高线的网格图形	path	设置或查询 MATLAB 的路径
meshgrid	用 x，y 网格数据	paren	各种括号的查询信息
meshz	带有零平面的三维网格图形	pascal	生成 Pascal 矩阵
min	求向量中最小元素	pause	暂停函数
more	控制命令窗口的输出页面	pcolor	伪颜色绘图
movie	播放存储的“电影”幅面	peaks	两变量的峰值函数演示
moviein	初始化“电影”各幅图象内存	penny	便士硬币的各个角度视图
mu2lin	声音文件对线性标度文件的转换	pi	圆周率(π)
NaN	不定值	pink	粉色色调颜色表
nargchk	函数输入输出参数个数检验	pinv	伪逆矩阵
nargin	函数中实际输入变量个数	plot	线性坐标图形绘制
nargout	函数中实际输出变量个数	plot3	绘制三维线或点型图形
newplot	Nextpolt 特性的. M 文件前缀	polar	极坐标图形绘制
nextpow2	找出下一个 2 的指数	poly	求矩阵的特征多项式

(续表)

函数名	含义	函数名	含义
polyder	多项式求导	return	返回到主调函数的命令
polyfit	数据的多项式拟合	rgb2hsv	RGB 对 HSV 颜色的转换
polyval	多项式求值	rgbplot	绘制颜色图
polyvalm	多项式矩阵求值	roots	求多项式的根
print	打印图形或将图形存盘	rose	极坐标(角度)直方图绘制
printopt	建立打印机默认值	rosser	典型的对称矩阵特征值问题测试
prism	光谱颜色表	rot90	将矩阵元素旋转 90 度
prod	对向量中各元素求积	round	截取到最近的整数
punct	各种标点符号的查询信息	rref	矩阵的行阶梯型实现
qr	矩阵的正交三角化(QR)分解	rrefmovie	消元法解方程过程演示
qrdelete	QR 分解中删除一列	rsf2csf	实块对角阵转移复块对角阵
qrinsert	QR 分解中插入一列	save	将工作空间中变量存盘
quad	低阶数值积分算法	saxis	声音坐标轴处理
quad8	高阶数值积分算法	surfnorm	表面图形规范化
quaddemo	自适应变步长数值演示	schur	Schur 分解
quake	Loma Prieta 地震模型	script	MATLAB 语句及文件信息
quit	退出 MATLAB 环境	sec	正割函数
quiver	箭头图形	sech	双曲正割
qz	广义特征值问题求解(QZ 算法)	semilogx	x 轴半对数坐标图形绘制
rand	产生[0，1]均匀分布随机矩阵	semilogy	y 轴半对数坐标图形绘制
randn	产生标准正态分布随机阵	sepdemo	有限元网格图演示
randperm	随机置换向量	set	设置对象属性
rank	求矩阵的秩	setstr	将数值转换为字符串
rbbox	擦除框	shading	阴影模式
rcond	LINPACK 倒数条件估计	sigdemo1	离散 Fourier 变换演示
real	求取实部函数	sigdemo2	连续 Fourier 变换演示
realmax	最大浮点数值	sign	符号函数
realmin	最小浮点数值	sin	正弦函数
relop	各种关系符号的查询信息	sinh	双曲正弦
rem	除法的余数	size	查询矩阵的维数
reset	恢复对象特性	slash	求解线性方程(左除右除)信息

(续表)

函数名	含　　义	函数名	含　　义
slice	容量可视图形	str2mat	字符串转换成矩阵
sort	对向量中各元素排序	str2num	字符串转换为实型数据
sound	将数据向量转换为声音	strcmp	字符串比较
sounddemo	MATLAB4.0 的声音功能演示	strings	关于 MATLAB 字符串的帮助信息
spalloc	给非零元素定位存储空间	subplot	将图形窗口分成若干个区域
sparse	从常规矩阵转换稀疏矩阵	subscribe	成为 MATLAB 的签约用户
sparsity	稀疏矩阵排序效应演示	subspace	子空间
spaugment	建立最小二乘增广系统	sum	对向量中各元素求和
spconvert	由稀疏矩阵外部格式进行转换	sunspots	太阳黑子活动模拟
spdiags	稀疏对角矩阵	surf	三维表面图形
specular	反射	surface	创建曲面
speye	稀疏单位矩阵	surfc	带有等高线的三维表面图形
spfun	对稀疏矩阵处理的非线性函数	surfl	带有光照阴影的三维表面图形
sphere	产生球面	svd	奇异值分解(SVD)
spinmap	使颜色旋转	symbfact	符号因式分解
spline2d	二维样条函数演示	symmmd	对称最小阶次
spones	将原稀疏矩阵非零元素有 1 取代	symrcm	逆序 Cuthill-McKee 排序
spparms	设置稀疏矩阵参数	tan	正切函数
sprank	结构秩数	tanh	双曲正切
sprandn	稀疏随机矩阵	terminal	设置图形终端类型
sprandsym	稀疏对称随机矩阵	text	在图形上加文字说明
sprintf	按照 C 语言格式书写字符串	tic	启动秒表计时器
spy	绘制稀疏矩阵结构	title	给图形加标题
sqdemo	超二次锥面的显示	toc	续取秒表计时器
sqrt	平方根函数	toeplitz	生成 Toeplitz 矩阵
sqrtm	矩阵的平方根	trace	求矩阵的迹
sscanf	按照 C 语言格式读字符串	trapz	梯形法求数值积分
stairs	阶梯图形绘制	treelayout	树状结构
startup	MATLAB 自启动文件	treeplot	画出分割路径的图形
std	求向量中各元素标准方差	tril	提取矩阵的下三角部分
stem	离散序列柄状图形绘制	triu	提取矩阵的上三角部分

(续表)

函数名	含　　义	函数名	含　　义
type	列出 M 文件	what	列出当前目录下的有关文件
uicontrol	建立用户界面控制的函数	whatsnew	手册中未给出的新特性
uigetfile	标准读盘文件名处理对话框	which	找出函数与文件所在的目录名
uisetcolor	标准颜色设置对话框	while	循环语句
uisetfont	标准字体设置树对话框	whitebg	将图形窗口设置成白色背景
unix	执行操作系统命令并回结果	who	简要列出工作空间变量名
unwrap	除去每 360°的跳跃	whos	详细列出工作空间变量名
upper	将一个字符串内容转换为大写	why	给出简要的回答
vander	生成 Vandermonde 矩阵	wilkinson	生成 Wilkinson 特征值测试矩阵
ver	显示程序版本号	xlabel	给图形加 X 坐标说明
version	显示 MATLAB 版本号	xor	逻辑异或
vibes	L 型振荡动画	ylabel	给图形加 Y 坐标说明
view	三维图形视口指定	zerodemo	求换演示
viewmtx	显示坐标变换矩阵	zeros	产生零矩阵
waterfall	瀑布型图形	zlabel	给图形加 Y 坐标说明

附录 B　MATLAB 工具箱函数命令索引表

1. 符号数学工具箱(Symbolic Math Toolbox)

函数名	含　　义	函数名	含　　义
' .'	复共轭转置、实转置	fourier	Fourier 变换
.* ./ .\ .^	数组乘、数组右除、数组左除、数组乘方	funtool	函数计算器
		horner	多项式分层表示
+ - * / \ ^	加、减、乘、右除、左除、乘方	ifourier	Fourier 逆变换
		ilaplace	Laplace 逆变换
ccode	符号表达式 C 代码	imag	复数虚部
char	符号对象转化为字符串	int	积分
collect	合并同类项	inv	逆矩阵
colspace	矩阵列空间的基	iztrans	Z 逆变换
compose	复合函数	jacobian	Jacobi 矩阵
conj	复共轭	jordan	矩阵 Jordan 标准形
cosint	余弦积分 Ci(x)	lambertw	$\lambda(x)e^{\lambda(x)}=x$ 的解
det	行列式	laplace	Laplace 变换
diag	产生对角阵或取矩阵对角线	latex	符号表达式 LaTeX 表示
diff	求导函数	limit	极限
digits	设置计算字长	maple	连接 Maple 核
double	符号对象转化为数值	mapleinit	初始化 Maple
dsolve	微分方程的解	mfun	Maple 函数求值
eig	特征值和特征向量	mfunlist	Maple 函数列表
expand	多项式展开	mhelp	Maple 帮助
expm	矩阵指数函数	null	矩阵零空间的基
ezplot	函数图	numden	分式表示
factor	多项式因式分解	poly	矩阵特征多项式
findsym	找出函数自变量	poly2sym	系数向量转化为符号多项式
finverse	逆函数	pretty	符号表达式打印形式
fortran	符号表达式 Fortran 代码	procread	安装 Maple 过程

(续表)

函数名	含　　义	函数名	含　　义
rank	矩阵的秩	sym	转化为符号对象
real	复数实部	sym2poly	符号多项式转化为系数向量
rref	化为行最简形	syms	产生符号变量
rsums	积分 Riemann 和演示	symsum	级数和
simple	寻找最短表达式	taylor	Taylor 级数展开
simplify	化简	tril	下三角形
sinint	正弦积分 Ci(x)	triu	上三角形
solve	代数方程的解	vpa	指定字长计算
subexpr	用子表达式表示	zeta	Riemann Zeta 函数 sum(1/k^z, k, 1, inf)
subs	变量代换		
svd	奇异值分解	ztrans	Z 变换

2. 统计工具箱(Statistics Toolbox)

函数名	含　　义	函数名	含　　义
anova1	单因素方差分析	boxplot	箱图
anova2	双因素方差分析	capable	质量管理性能指标
barttest	Bartlett 试验	capaplot	质量管理性能图
betacdf	Beta 分布分布函数	cdf	分布函数
betafit	Beta 分布参数估计	chi2cdf	卡方分布分布函数
betainv	Beta 分布分位数	chi2inv	卡方分布分位数
batalike	Beta 分布对数似然函数	chi2pdf	卡方分布密度函数
betapdf	Beta 分布密度函数	chi2rnd	卡方分布随机数
betarnd	Beta 分布随机数	chi2stat	卡方分布均值和方差
betastat	Beta 分布均值和方差	cluster	树聚类
binocdf	二项分布分布函数	clusterdata	数据聚类
binofit	二项分布参数估计	cophenet	聚类相关系数
binoinv	二项分布分位数	cordaxch	使用坐标变换的行列式最优试验设计
binopdf	二项分布密度函数		
binornd	二项分布随机数	corrcoef	相关系数
binostat	二项分布均值和方差	cov	协方差阵
bootstrp	重新抽样	daugment	试验设计行列式优化分析

（续表）

函数名	含　　义	函数名	含　　义
dcovary	固定协方差的行列式最优试验设计	gline	交互画直线
dendrogram	聚类分层递阶树图	gname	交互标记点坐标
errorbar	误差图	hadamard	Hadamard 设计
ewmaplot	指数权移动平均图	harmmean	调和平均
expcdf	指数分布分布函数	histfit	直方图和正态密度曲线拟合
expfit	指数分布参数估计	hygecdf	超几何分布分布函数
expinv	指数分布分位数	hygeinv	超几何分布分位数
exppdf	指数分布密度函数	hygepdf	超几何分布密度函数
exprnd	指数分布随机数	hygernd	超几何分布随机数
expstat	指数分布均值和方差	hygestat	超几何分布均值和方差
fcdf	F 分布分布函数	icdf	分位数
ff2n	两水平全因子设计	inconsistent	聚类相容性
finv	F 分布分位数	iqr	中部 50%数据极差
fpdf	F 分布密度函数	kurtosis	峰度
frnd	F 分布随机数	linkage	聚类树
fstat	F 分布均值和方差	logncdf	对数正态分布分布函数
fsurfht	交互函数等高线图	logninv	对数正态分布分位数
fullfact	混合水平全因子设计	lognpdf	对数正态分布密度函数
gamcdf	gamma 分布分布函数	lognrnd	对数正态分布随机数
gamfit	gamma 分布参数估计	lognstat	对数正态分布均值和方差
gaminv	gamma 分布分位数	lscov	给定协方差阵的回归
gamlike	gamma 分布对数似然函数	lsline	最小二乘拟合直线
gampdf	gamma 分布密度函数	mad	平均绝对偏差
gamrnd	gamma 分布随机数	mean	均值
gamstat	gamma 分布均值和方差	median	中值
geocdf	几何分布分布函数	mle	极大似然估计
geoinv	几何分布分位数	moment	各阶中心矩
geomean	几何平均	nanmax	忽略 NaN 的最大值
geopdf	几何分布密度函数	nanmean	忽略 NaN 的均值
geornd	几何分布随机数	nanmedian	忽略 NaN 的中值
geostat	几何分布均值和方差	nanmin	忽略 NaN 的最小值

（续表）

函数名	含　　义	函数名	含　　义
nanstd	忽略 NaN 的标准差	normpdf	正态分布密度函数
nansum	忽略 NaN 的和	normplot	正态分布检验图
nbincdf	负二项分布分布函数	normrnd	正态分布随机数
nbininv	负二项分布分位数	normspec	正态密度曲线区间图
nbinpdf	负二项分布密度函数	normstat	正态分布均值和方差
nbinrnd	负二项分布随机数	pareto	Pareto 统计图
nbinstat	负二项分布均值和方差	pcacov	协方差主成分分析
ncfcdf	非中心 F 分布分布函数	pcares	主成分分析残差
ncfinv	非中心 F 分布分位数	pdf	密度函数
ncfpdf	非中心 F 分布密度函数	pdist	对距离
ncfrnd	非中心 F 分布随机数	poisscdf	Poisson 分布分布函数
ncfstat	非中心 F 分布均值和方差	poissfit	Poisson 分布参数估计
nctcdf	非中心 t 分布分布函数	poissinv	Poisson 分布分位数
nctinv	非中心 t 分布分位数	poisspdf	Poisson 分布密度函数
nctpdf	非中心 t 分布密度函数	poissrnd	Poisson 分布随机数
nctrnd	非中心 t 分布随机数	poissstat	Poisson 分布均值和方差
nctstat	非中心 t 分布均值和方差	polyconf	多项式预测
ncx2cdf	非中心卡方分布分布函数	polyfit	多项式拟合
ncx2inv	非中心卡方分布分位数	polyval	多项式取值
ncx2pdf	非中心卡方分布密度函数	prctile	百分位数
ncx2rnd	非中心卡方分布随机数	princomp	行数据矩阵主成分分析
ncx2stat	非中心卡方分布均值和方差	qqplot	分布类型比较图
nlinfit	非线性最小二乘拟合	random	随机数
nlintool	非线性拟合预测图	range	极差
nlparci	非线性拟合参数置信区间	ranksum	秩和检验
nlpredci	非线性拟合预测置信区间	raylcdf	Rayleigh 分布分布函数
nnls	非负最小二乘法	raylinv	Rayleigh 分布分位数
normcdf	正态分布分布函数	raylpdf	Rayleigh 分布密度函数
normfit	正态分布参数估计	raylrnd	Rayleigh 分布随机数
norminv	正态分布分位数	raylstat	Rayleigh 分布均值和方差
normlike	正态分布对数似然函数	rcoplot	回归分析残差图

（续表）

函数名	含　　义	函数名	含　　义
refcurve	参考多项式曲线	unidcdf	离散均匀分布分布函散
refline	参考直线	unidinv	离散均匀分布分位数
regress	多元线性回归	unidpdf	离散均匀分布密度函数
ridge	岭回归	unidrnd	离散均匀分布随机数
rowexch	使用行变换的行列式最优试验设计	unidstat	离散均匀分布均值和方差
rstool	回归响应工具	unifcdf	均匀分布分布函数
schart	标准差时间图	unifinv	均匀分布分位数
signrank	符号秩检验	unifit	均匀分布参数估计
signtest	成对样本符号检验	unifpdf	均匀分布密度函数
skewness	偏度	unifrnd	均匀分布随机数
squareform	将 pdist 输出转化为方阵	unifstat	均匀分布均值和方差
std	标准差	var	方差
stepwise	逐步回归用户界面	weibcdf	Weibull 分布分布函数
surfht	交互等高线图	weibinv	Weibull 分布分位数
tcdf	t 分布分布函数	weibpdf	Weibull 分布密度函数
tinv	t 分布分位数	weibplot	Weibull 分布检验随
tpdf	t 分布密度函数	weibrnd	Weibull 分布随机数
trimmean	忽略异常数据的均值	weibstat	Weibull 分布均值和方差
trnd	t 分布随机数	xbarplot	均值时间图
tstat	t 分布均值和方差	zscore	计算距离前的规范化
ttest	单样本 t 检验	ztest	单样本 z 检验
ttest2	双样本 t 检验		

3. 样条工具箱(Spline Toolbox)

函数名	含　　义	函数名	含　　义
fnval	样条函数求值	fn2fm	样条形式转换
fnbrk	样条函数各部分	fnder	样条函数求导
fncmb	样条函数运算	fnint	样条函数积分
spline	非扭结端点三次样条插值	fnjmp	样条函数跳 $f(x+)-f(x-)$
mkpp	形成样条 pp 形式	fnplt	样条函数图
unmkpp	展开样条 pp 形式	fnrfn	添加划分分点

（续表）

函数名	含　　义	函数名	含　　义
csapi	非扭结端点三次样条插值	newknt	新节点分布
csape	各类端点三次样条插值	optknt	最优节点
csaps	三次样条拟合	slvblk	求解几乎块对角方程组
cscvn	自然或周期三次样条	bkbrk	拆分几乎块对角方程组
getcurve	三次样条交互产生	spdemos	样条值示命令
pplst	pp 样条命令列表	splexmpl	几个样条例子
ppbrk	pp 样条分解	ppalldm2	pp 样条简介
ppmak	pp 形式样条构造	spalldm2	*B* 样条简介
pprfn	pp 样条加入节点	bspline	*B* 样条图
ppual	pp 形式样条函数值	bsplidem	几个 *B* 样条演示
splst	*B* 样条命令列表	csapidem	三次样条插值演示
spbrk	*B* 样条分解	csapsdem	三次样条拟合演示
spmak	*B* 形式样条	pckkntdm	节点选取演示
sprfn	*B* 样条加入节点	spcrvdem	产生 *B* 样条曲线演示
spcrv	产生 *B* 样条曲线	difeqdem	一个奇异摄动常微分方程
spapi	*B* 样条插值	chebdem	等振荡样条
spap2	*B* 样条最小二乘拟合	tspdem	一个例子
spaps	*B* 样条拟合	franke	Franke 双变量试验函数
spcol	*B* 样条配置矩阵	subplus	截幂函数
augknt	分划和节点	titanium	钛温度数据
aveknt	节点平均	splpp	*B* 样条左半轴化为 pp 样条
brk2knt	分划及重数到节点求法	sprpp	*B* 样条右半轴化为 pp 样条
knt2mlt	节点到分划及重数求法	pp2sp	pp 形式样条转化为 *B* 样条
sorted	查询网格点	sp2pp	*B* 形式样条转化为 pp 样条

4. 最优化工具箱(Optimization Toolbox)

- MATLAB5.2 使用的优化工具箱(Optimization Toolbox 1.5.2)

函　数	功　　能	含　　义
attgoal	目标规划求解	$\min\limits_{x,\gamma} \gamma \quad s.t. \quad F(x) - w\gamma \leqslant \text{goal}$
conls	约束线性最小二乘法求解	$\min\limits_{x} \lVert C \cdot x - d \rVert_2^2 \quad s.t. \quad A \cdot x \leqslant b$ $\text{lb} \leqslant x \leqslant \text{ub}$

（续表）

函　　数	功　　能	含　　义
constr	约束线性规划求解	$\min_x f(x)$ s.t. $G(x) \leqslant \mathbf{0}$
curvefit	非线性曲线拟合	$\min_x \frac{1}{2} \| F(x, \mathbf{xdata}) - \mathbf{ydata} \|_2^2$ s.t. $\mathbf{lb} \leqslant x \leqslant \mathbf{ub}$
fmin	一元函数极小值求解	$\min_x f(x)$ s.t. $x_1 < x < x_2$
fmins, fminu	无约束非线性规划求解	$\min_x f(x)$
fsolve	多变量非线性方程组求解	$f(x)=\mathbf{0}$
fzero	单变量非线性方程求解	$f(x)=0$
leastsq	非线性最小二乘法求解	$\min_x \frac{1}{2} \| F(x) \|_2^2 = \frac{1}{2} \sum_i F_i(x)^2$ s.t. $\mathbf{lb} \leqslant x \leqslant \mathbf{ub}$
lp	线性规划求解	$\min_x \mathbf{f}'x$ s.t. $A \cdot x \leqslant b$ $\mathbf{lb} \leqslant x \leqslant \mathbf{ub}$
minmax	最大最小问题求解	$\min_{G(x) \leqslant 0} \max_{\{F_i\}} \{F_i(x)\}$
nnls	非负线性最小二乘法 X 求解	$\min_x \| C \cdot x - d \|_2^2$ s.t. $x \geqslant \mathbf{0}$
qp	二次规划求解	$\min_x \frac{1}{2} x'Hx + \mathbf{f}'x$ s.t. $A \cdot x \leqslant b$ $\mathbf{lb} \leqslant x \leqslant \mathbf{ub}$
seminf	半无穷条件下的非线性规划求解	$\min_x \mathbf{f}(x)$ s.t. $K(x, w) \leqslant \mathbf{0}$ 对所有的 w 成立

- MATLAB5.3 使用的优化工具箱(Optimization Toolbox 2.0 兼容 1.5.2)

函　　数	功　　能	含　　义
fminbnd	一元函数极小值求解	$\min_x \mathbf{f}(x)$ s.t. $x_1 < x < x_2$
fminunc, fminsearch	无约束非线性规划求解	$\min_x \mathbf{f}(x)$
linprog	线性规划求解	$\min_x \mathbf{f}'x$ s.t. $A \cdot x \leqslant b$ $Aeq \cdot x = \mathbf{beq}$, $\mathbf{lb} \leqslant x \leqslant \mathbf{ub}$
quadprog	二次规划求解	$\min_x \frac{1}{2} x'Hx + \mathbf{f}'x$ s.t. $A \cdot x \leqslant b$ $Aeq \cdot x = \mathbf{beq}$, $\mathbf{lb} \leqslant x \leqslant \mathbf{ub}$
fmincon	约束线性规划求解	$\min_x f(x)$ s.t. $c(x) \leqslant \mathbf{0}$, $\mathbf{ceq}(x) = \mathbf{0}$ $A \cdot x \leqslant b$, $Aeq \cdot x = \mathbf{beq}$, $\mathbf{lb} \leqslant x \leqslant \mathbf{ub}$

（续表）

函　　数	功　　能	含　　义
fgoalattain	目标规划求解	$\min_{x,\gamma} \gamma$　s. t.　$F(x)-w\gamma \leqslant \text{goal}$ $c(x)\leqslant 0$, $\text{ceq}(x)=0$, $A\cdot x\leqslant b$ $\text{Aeq}\cdot x=\text{beq}$, $\text{lb}\leqslant x\leqslant \text{ub}$
fminmax	最大最小问题求解	$\min_{x}\max_{\{F_i\}}\{F_i(x)\}$ s. t.　$c(x)\leqslant 0$, $\text{ceq}(x)=0$ $A\cdot x\leqslant b$, $\text{Aeq}\cdot x=\text{beq}$, $\text{lb}\leqslant x\leqslant \text{ub}$
fseminf	半无穷条件下的非线性规划求解	$\min_{x} f(x)$　s. t.　$K(x, w)\leqslant 0$ 对所有 w, $c(x)\leqslant 0$, $\text{ceq}(x)=0$ $A\cdot x\leqslant b$, $\text{Aeq}\cdot x=\text{beq}$, $\text{lb}\leqslant x\leqslant \text{ub}$
fzero	单变量非线性方程求解	$f(x)=0$
fsolve	多变量非线性方程组求解	$f(x)=0$
lsqnonneq	非负线性最小二乘法求解	$\min_{x}\Vert C\cdot x-d\Vert_2^2$　s. t.　$x\geqslant 0$
lsqlin	约束线性最小二乘法求解	$\min_{x}\Vert C\cdot x-d\Vert_2^2$　s. t.　$A\cdot x\leqslant b$ $\text{Aeq}\cdot x=\text{beq}$, $\text{lb}\leqslant x\leqslant \text{ub}$
lsqnonlin	非线性最小二乘法求解	$\min_{x}\frac{1}{2}\Vert F(x)\Vert_2^2=\frac{1}{2}\sum_i F_i(x)^2$ s. t.　$\text{lb}\leqslant x\leqslant \text{ub}$
lsqcurvefit	非线性曲线拟合	$\min_{x}\frac{1}{2}\Vert F(x, \text{xdata})-\text{ydata}\Vert_2^2$ s. t.　$\text{lb}\leqslant x\leqslant \text{ub}$

习题参考答案

实 验 一

1. (1) $x_1 = 2.3830$，$x_2 = 1.4894$，$x_3 = 2.0213$；

   (2) $x_1 = -0.4706$，$x_2 = -0.2941$，$x_3 = 0$；

   (3) $x_1 = 0.3311$，$x_2 = -0.1219$（超定最小二乘解）；

   (4) 无穷多解。$x_1 = -1.5k + 1$，$x_2 = 1.5k$，$x_3 = 0.5k + 1$；$x_4 = k$。

2. 一年(24%，76%)，两年(27%，73%)，十年(49%，51%)，最终(83%，17%)。

3. (37.57，25.79，24.77)。

4. $a = 13.3$，$b = 6.4$，$c = 8.5$。

5. $f = 2y_1^2 + 2y_2^2 + 7y_3^2$，变换矩阵 $C = \begin{pmatrix} -0.8433 & -0.4216 & -0.3333 \\ -0.1054 & 0.7379 & -0.6667 \\ -0.5270 & 0.5270 & 0.6667 \end{pmatrix}$

7. 不选主元：(1，1)，(0.9992，1)，(2.2204，1)；

   选主元：均为(1，1)。

8. 条件数 1.7636e+016。

实 验 二

3. 624000，187200，3025。

6. 先作图观察交点，再用 fsolve 求解。(1.7362，-2.6929)，(1.6581，1.8936)，(3.4829，-5.6394)，(4.0287，-4.1171)；

7. 0.2968。

10. 3.4495，3.5441。

实 验 三

2. 上升：[-2，-1.6926]，[-1.2401，1.2401]，[1.6926，2]；

   下降：[-1.6926，-1.2401]，[1.2401，1.6926]。

5. (1) 极大值点：-1.5326，0；极小值点：-0.7315，1.5951。

   (2) 极大值点-2，极小值点 2。

   (3) 极大值点 1，极小值点 2。

6. 20.8097m。

8. 极大值点(-1/3，-6)，极小值点(0，0)，鞍点(-7/6，-7/2)，(5/6，-5/2)。

9. 1/2，1/4，1/6，……。

10. $1/a + (c_0 - k\ln M)/(1 - ak)$。

13. 极大值点：-1.6926，1.2401，2；极小值点：-2，-1.2401，1.6926。

实 验 四

2. 0.3413，0.9460，1.2912，35843.789。
3. 15.8651。
4. (1) 6.1879；(2) 3.1383。
5. 0(注意被积函数为奇函数)。
6. $2k(k = 4, 6, 8)$。
8. 291.8696。
9. 178.5。
10. 26.3452。
11. 18400 万元。
12. 盈余 $35m^2$。
14. 1.3433，0.9063，3.1044。

实 验 五

3. $y' = 2x + y^2$，$y(0) = 0$。求数值解。
5. $T' = k(c-T)$，$T(0) = 20$、$T(10) = 25.2$、$T(20) = 28.52$。解得户外温度 $c = 33$，比例系数 $k = 0.05$。
6. (2) 8.65。
7. (1) $V_{max} = \exp(20)$；(2) $t = 0.353$；(3) $t = 30$；(4) 451，0.4，9.6
8. $20 - 20\exp(-5000t)[\cos(5000t) + \sin(5000t)]$。

实 验 六

5. 0.193，20.8，0.366。
6. 0.8119。极大值点：(−0.3827，−0.9239)，(−0.3827，−0.9239)。

实 验 七

2. 0.7373。
5. 否，$U = 1.5607 < 1.6449$；是，$U = 4.264 > 1.6449$。方法参见习题 4。
6. 否，$U = 1.2772 < 1.6449$。方法参见习题 4。
7. (1) 是，$U = 2.6756 > 1.6449$。(2) 是，$U = 1.9781 > 1.6449$。
8. 设收入 x_1，价格 x_2，则需求 $y = 111.6918 + 0.0143x_1 - 7.1882x_2$。预测值 82.8594。
9. $y = 9.7966 + 5.4160x_1 + 12.5843x_2$（去掉一个异常点后）。
10. 否，$\chi^2 = 5.8641 < 9.4877$。
11. 否。
12. 线性 $66.5176 + 0.4139x_1 - 0.2698x_2$，预测值 94。

实 验 八

1. 298.6，322.2。
6. $v_0 = 5.6221$，$\tau = 3.5269$。
7. 二次曲线 $-0.0592x^2 + 2.3265x - 0.9803$
8. $y = 111.44 - 9.03/x$。
9. 分段三次式系数

(1) 1.8863	−1.0143	1.0000	0.5000
0.7952	−0.7314	0.9127	0.5477
0.6320	−0.5167	0.8004	0.6245
0.3151	−0.4029	0.7452	0.6708
(2) −6.2652	0.0000	0.9697	0.5000
1.8813	−0.9398	0.9227	0.5477
−0.4600	−0.4318	0.7992	0.6245
2.1442	−0.5146	0.7424	0.6708

11. 用曲面样条插值，步长10米，最高点(170，180)，高程721。

实 验 九

1. $x=[0, 8, 0, -6]$，$f=2$。
2. $x=[0, 3, 0]$，$f=12$。
3. $x=[20, 24]$，$f=428$。
4. $x=[0. 0.6667, 0.3333]$，$f=7.2333e+003$。
5. $x=[0, 4, 0, 0, 2, 0]$，$f=-16.0$。

实 验 十

1. $x=[-0.0001, -1.0000]$，$f=4.0196e-015$。
2. $x=1.0e-004*[0.0722, -0.3580]$，$f_2=5.9959e-009$。
3. $x=1.0e-004*[0.1860, 0.0808]$，$f=40$。
4. $f=43.0858$，$x=[86.2903, 103.7573, 127.1007, 152.1905]$。
5. $x=[22.5850, 12.5850, 12.1225]$，$f=3.4456e+003$。
6. $x=[49,9999, 60.0003, 69.9998]$，$f=1280e+004$。

实 验 十 一

1. $x=[1, 2]$，$f=1$。
2. $x=[0, 0, 1, 2]$，$z=88$。
3. $x=[1, 2, 3]$，$f=29$。
4. $x=[0, 0, 0, 0, 1, 0, 1]$，$f=44$。
5. $x=[0, 0, 1]$，$f=2$。
6. $x=[0, 0, 0, 1]$，$f=4$。

实 验 十 二

1. 1、3年购置新设备，或1、4年购置新设备，总支付费用均53。
2. 按顶点序号：$a-b$；$a-b-e$；$a-b-e-i$；$a-b-e-i-g$；$a-c$；$a-b-e-i-g-j$；$a-b-e-i-h$；$a-c-d$；$a-b-e-i-h-k$；$a-f$；

   按上面的次序最短距离分别为：0　2　3　4　7　8　8　11　15　20　∞。
3. 两条路线均为$a-b-d-f$，距高9个单位，需时20个单位。
4. 10.2海里。

实 验 十 三

1. 第一、二区各设2个点，获利1436。

2. 最短路径为 1—4—6—8—10,距离为 14。

3. 第一、二、三季度生产 2 艘,第四季度生产 1 艘。

参 考 文 献

[1] 薛定宇. 控制系统计算机辅助设计——MATLAB语言及应用. 北京:清华大学出版社,1996
[2] 李人厚等. 精通 MATLAB综合辅导与指南. 西安:西安交通大学出版社,1998
[3] 程卫国等. MATLAB5.3应用指南. 北京:人民邮电出版社,1999
[4] 张宜华. 精通 MATLAB5. 北京:清华大学出版社,1999
[5] 姜健飞等. 线性代数. 上海:中国纺织大学出版社,1999
[6] Ecker. *Studio Calculus*. New York: Harper Collins College Publisher,1996
[7] 同济大学编. 高等数学(上、下). 北京:高等教育出版社,1988
[8] 中山大学编. 常微分方程. 北京:人民教育出版社,1978
[9] 冯康等. 数值计算方法. 北京:国防工业出版社,1978
[10] Nakamura S. *Numerical Analysis and Graphic Visualization*. London: Prentice Hall. 1996
[11] Zachary. *Introduction to Scientific Programming: Computational Problems solving using Maple and C*. New York: Springer,1996
[12] 郑颖等. 概率论与数理统计. 上海:中国纺织大学出版社,1999
[13] 施锡铨等. 数据分析方法. 上海:上海财经大学出版社,1997
[14] 叶其孝. 大学生数学建模竞赛辅导教材(二). 长沙:湖南教育出版社,1997
[15] 寿纪麟. 数学建模方法与范例. 西安:西安交通大学出版社,1993
[16] 施光燕等. 最优化方法. 北京:高等教育出版社,1999
[17] 清华大学《运筹学》教材编写组. 运筹学. 北京:清华大学出版社,1995
[18] 清华大学《现代应用数学手册》编委会. 现代应用数学手册. 北京:清华大学出版社,1997
[19] 袁振东等. 数学建模. 上海:华东师大出版社,1997
[20] 谭永基等. 数学建模. 上海:复旦大学出版社,1997
[21] 萧树铁等. 数学实验. 北京:高等教育出版社,1999
[22] 乐经良等. 数学实验. 北京:高等教育出版社,1999